Pulse Shape Correlation for Laser Detection and Ranging (LADAR)

THESIS

Brian T. Deas, Major, USAF

AFIT/GE/ENG/10-07

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/10-07

Pulse Shape Correlation for Laser Detection and Ranging (LADAR)

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Brian T. Deas, BS

Major, USAF

March 2010

AFIT/GE/ENG/10-07

Pulse Shape Correlation for Laser Detection and Ranging (LADAR)

Brian T. Deas, BS
Major, USAF

Approved:

/signed/	25 February 2010
Dr. Stephen Cain, PhD (Chairman)	Date
/signed/	25 February 2010
Dr. Richard Martin, PhD (Member)	Date
/signed/	25 February 2010
Mr. Ernest Armstrong, (Member)	Date

Abstract

An un-biased range estimation algorithm was developed that countered errors caused by pulse shape effects in laser radar (LADAR) systems. The Advanced Scientific Concepts FLASH LADAR created three-dimensional images by measuring return intensity and time of flight for a single laser pulse. A range estimate was made by estimating the return time of flight from the peak amplitude of the return laser pulse. Range error estimates occurred when the shape of the measured return pulse deviated from the expected pulse shape.

A normalized variable shaping correlation algorithm called NOVAS performed the correlation between the recorded waveforms and a robust set of Gaussian-based reference waveforms covering a wide range of pulse shapes and asymmetries. While NOVAS did not generate the most accurate range estimates due to noise sensitivity, it was able to produce an estimate of the average asymmetric pulse shape. The NOVAS generated average (NGA) asymmetric pulse shape produced a more accurate average range estimate than the NGA symmetric pulse shape in six of seven trials and produced a more accurate average range estimate with less variance than an arbitrarily chosen symmetric reference waveform in all of seven trials.

Acknowledgements

My greatest of thanks go to Dr. Stephen Cain whose expertise and guidance facilitated my progress at every step along the way. His earnest concern for my success was evident in the effort and time he was willing to put forth to ensure I was able to meet and exceed my goals, as well as his encouragement in this endeavor.

I would also like to thank team "Walker Ranger": Maj Ryan "Lag" Howland, Mr. Kristofer "BIPOL" Peterson, Capt Reid "KEE-ROK" Larson, Ph.D, and Capt Sarah "Token" Helms. Each of them brought a key talent to the team without which we would not have even achieved a fraction of the success we reached. Thanks to the rest of the "TPS 09A Chucks" for the support of all to get through the year.

Brian T. Deas

Table of Contents

Page

List of Figures

List of Tables

List of Abbreviations

Abbreviation		Page
ASC	Advanced Scientific Concepts	1
LADAR	laser detection and ranging	1
NOVAS	normalized variable shaping	1
RPV	remotely piloted vehicle	1
AAR	air-to-air refueling	1
DGPS	differential global positioning system	2
APD	avalanche photo diodes	2
SULAR	staring under-water laser ranging	3
AFIT	Air Force Institution of Technology	5
TPS	Test Pilot School	5
TMP	test management project	5
RASCAL	reconfigurable airborne sensor, communication and laser	5
SNR	signal to noise ratio	9
MLE	maximum likelihood estimator	9
PDF	probability distribution function	9
FOV	field of view	38
NGA	NOVAS generated average	55
RMSE	root mean squared error	64

Pulse Shape Correlation for Laser Detection and Ranging (LADAR)

I. Introduction

1.1 Problem Description

This thesis reports the results of research and testing performed to characterize and improve range estimation accuracy of the Advanced Scientific Concepts (ASC) FLASH laser detection and ranging (LADAR) system. The objective is to describe the primary sources of range estimation error and to show how a new normalized variable shaping (NOVAS) correlation range estimation reduces or eliminates some of the error. Improving LADAR range estimation increases the feasibility of future Air Force applications such as automated air-to-air refueling and three dimensional target identification.

1.2 LADAR Application: Automated Air-to-Air Refueling

The LADAR system possesses capabilities potentially useful for bridging a critical gap in the progress of the remotely piloted vehicle (RPV). The movement of airpower towards RPVs has required technology to replicate or improve upon human-performed functions. One critical element of global airpower is the ability to refuel aircraft in midair, which could eliminate fuel as a limiting factor in the duration of that aircraft's mission. With the removal of the human crew limitations, an RPV with the ability to perform autonomous air-to-air refueling (AAR) would be able to sustain its mission indefinitely.

Completing this task requires systems that can aid in a critical skill element of AAR, the ability to precisely maintain relative positioning between the refueler and the receiver aircraft while in close formation. Until the refueling boom makes contact,

the pilot of the receiver aircraft maintains the responsibility of visually maneuvering to and maintaining the pre-contact and contact positions. The human eye is the primary sensor in this system.

The ability to perform the close formation task required for AAR has been demonstrated using a differential global positioning system (DGPS) as the sensor. The system was able to generate an average range accuracy of 1.9 cm and average spherical position error of 3.3 cm during 11 hours of close formation flight [11]. However GPS signals can be jammed, affected by natural events such as solar flares, or completely denied by disrupting or degrading the GPS satellite constellation. Therefore non–GPS solutions to autonomous AAR are sought to provide this capability in a GPS denied environment.

The ASC FLASH LADAR possesses characteristics well suited for replicating this function. The ASC FLASH LADAR combines the unique ability to image and range a three dimensional target scene simultaneously without scanning or boresighting the laser or reliance upon GPS. In a series of rapid snapshots, the system could potentially provide a three-dimensional image from which an RPV would be able to determine all the critical elements of maintaining formation: relative range, closure, angle, and angle rate between the two aircraft.

1.3 LADAR Operation

The ASC FLASH LADAR system operates by sending out a single pulses of laser light generated from a Q-switched laser cavity. A detector then measures and records the intensity of the reflected return pulse as well as the round trip time of flight. The data output can be used to generate a three dimensional image of the target scene.

The system uses an array of avalanche photo diode (APD) detectors to create a 128×128 image from a single laser pulse [7]. Each detector in the array has the capability of recording 20 laser return intensity values. In a given instant, the

photons of laser light falling upon each single detector are amplified by the APD and output as a voltage, which is stored temporarily in one of 20 locations allotted to that detector [9]. These locations are referred to as *slices* in this thesis in that they are essentially time slices of the returning laser pulse. Each of the 20 slices are filled sequentially, with the sequencing being controlled by an electronic oscillator operating at a nominal frequency of 400 MHz. As time passes, each detector outputs values that cycle through the slices, overwriting values that occurred 20 slices ago but keeping track of the total number of slices recorded. At any given time, there is a 20 slice time history stored that is updated approximately every 2.5 ns.

The LADAR stops overwriting previous data based on the mode of operation. In stop-mode, a particular pixel stops overwriting when the detector for that pixel outputs a voltage above a pre-defined threshold. When this threshold is exceeded in a given slice, that slice, the previous 12 slices and the subsequent six slices are saved. The thirteenth previous slice is set to zero value and is called the marker slice. The LADAR permanently stores the twenty slices as well as the total number slices that have passed at the time the marker slice is saved. Slices are saved in their original stored position in the sequence of 20 and are circularly wrapped around the sequence.

The second mode of operation was designed for operation with visual obscuration such as haze, smoke, or water and is called staring under-water laser ranging (SULAR). To prevent early triggering from the obscuration, all the pixels trigger simultaneously at a user defined length of time. The approximate range to the target must be known for this mode to be useful. The method of recording the data is identical in both modes [1].

Figure 1 shows an actual LADAR return as it was recorded by the system. First, the LADAR stored values in slices 1-20, circularly overwriting the previous values multiple times as time passed. Then slice four exceeded the threshold value causing the system to trigger. Slice 12 was set to zero as the marker slice and the slice count to this point was saved. Slices 13-20 and 1-3 were saved preventing them

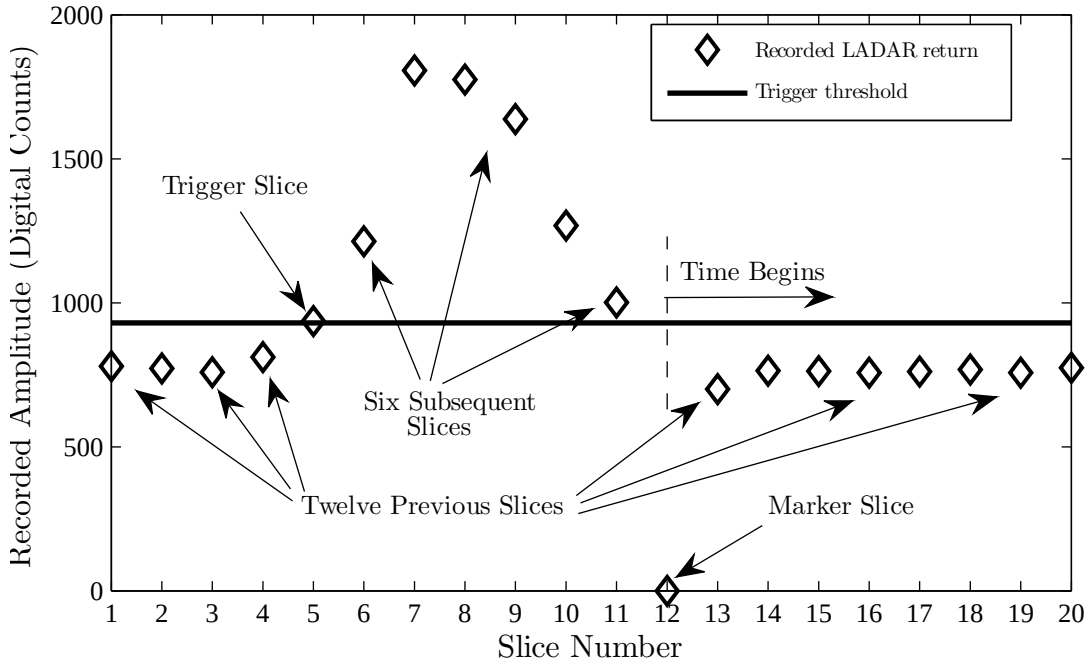

Figure 1: An actual unaltered LADAR waveform

from being overwritten. Finally, slices 5-11 were saved sequentially as they occurred, completing the final set of 20 filled slices. These values were then saved in an output file and the slices were cleared in preparation for reception of the next outgoing laser pulse.

A critical element in accurate ranging is precise determination of the true round trip time-of-flight of the laser pulse. The range to the target, R, is related to this round trip time, t_{rt}, by the simple relationship to the speed of light, c, given by:

$$R = \frac{t_{rt} \cdot c}{2}. \tag{1}$$

For this analysis, the true time-of-flight will be defined by the time interval between the transmission of the peak of the outgoing pulse to the reception of the peak amplitude of the return pulse. This problem is confounded by the sampling interval of

the system. As can be seen in Figure 1, the true peak of the return pulse lies between sampling times, and therefore has to be estimated.

1.4 Overview of Discussion

The ASC FLASH LADAR operates without the aid of GPS and therefore is not vulnerable to the disruptive effects mentioned above. However there are limitations in the system that prevent range measurements to be as accurate as those from the DGPS system. While some of the limitations are related to hardware and task implementation of the particular system used in this research, there are also limitations associated with the methods used to process the LADAR signal and generate the range estimates. Chapter II discusses in detail the sources of these errors as well as previous work performed in this area of research.

Chapter III introduces the NOVAS algorithm. This algorithm possesses advantageous characteristics over other common range estimation techniques and applies to many LADAR systems and applications. Chapter III also includes discussion of NOVAS implementation in simulation which was performed at the Air Force Institute of Technology (AFIT) in preparation for ground testing at the Air Force Test Pilot School (TPS). Simulation showed that NOVAS had the capability to generate more accurate range estimates than non-adaptive, fixed reference waveforms.

Chapter IV discusses the ground testing which occurred in conjunction with the *Walker Ranger* test management project (TMP). Ground testing was performed in accordance with the *Walker Ranger* test plan and took place at TPS with the LADAR loaded inside the reconfigurable airborne sensor, communication and laser (RASCAL) pod. The RASCAL pod was mounted externally on the wing station of a Lockheed Martin F-16D aircraft in preparation for future flight testing.

Chapter IV also discusses the results of the ground testing. It was shown that the laser pulses recorded by the LADAR had significant variability to the pulse shapes. It was also shown that NOVAS was able to eliminate some of the error due to the sources

discussed in Chapter II while adapting to the variation in return pulse shapes. While NOVAS did not generate the most accurate range estimates due to noise sensitivity, it was able to produce an estimate of the average asymmetric pulse shape. The NOVAS generated average (NGA) *asymmetric* pulse shape produced a more accurate average range estimate than the NGA *symmetric* pulse shape in six of seven trials.

Chapter V further demonstrates that the NGA asymmetric pulse shape produced a more accurate average range estimate with less variance than an *arbitrarily chosen symmetric* reference waveform in all of seven trials. Chapter V concludes that the benefit of the NOVAS algorithm was the ability to accurately estimate the average pulse shape which was then used to generate more accurate and more precise range estimates amidst large sources of noise within the system.

II. Background Theory

2.1 LADAR Pulse Formation

The FLASH LADAR system uses a Q-switched laser to create the outgoing pulse. A Q-switched laser creates a laser pulse by pumping the gain medium to a level above threshold while spoiling the cavity oscillation. Oscillation conditions are then restored for a short period allowing the energy stored in the gain medium to be extracted in one large pulse. Siegman holds that the two main contributors to the pulse shape are the initial inversion ratio $r = N_i/N_{th}$, where N_i is the number electrons in the excited state and N_{th} is the threshold number of electrons required to create a laser, and the empty cavity photon lifetime, τ_c [10]. The instant cavity oscillation conditions are restored, the rate of change of the number of photons oscillating in the cavity, $n(t)$, is given by:

$$\frac{dn(t)}{dt} = \frac{(r-1)}{\tau_c} n(t). \tag{2}$$

This holds given the assumption that during the pulse buildup time the population inversion remains unchanged [10]. This gives the solution:

$$n(t) = n_i e^{\frac{(r-1)t}{\tau_c}}, \tag{3}$$

describing an exponential ramp–up of the pulse where n_i is the initial number of excited electrons. At this point, the pulse depletes the population inversion to a level below threshold, and the pulse exponentially decays with a dependence on τ_c and r described by coupled rate equations [10]. A cavity with a large r will ramp up more quickly causing the pulse shape to be more asymmetric while a pulse with a lower value of r will have a more symmetric shape [10].

2.2 Modeling of Q-switched Laser Pulses

Liu describes a process for modeling a Q-switched pulse without solving the coupled rate equations mentioned above. Liu holds that parameters of the pulse can be gathered from knowledge of the cavity structure, the system gains and losses,

the inversion at maximum photon density, and the final inversion at zero photon density [5]. These parameters contain sufficient information to fully describe the wavefront envelope of the pulse. The largest contribution by Liu is the pulse symmetry factor, ϵ. This can be found for a threshold inversion ratio, z_t and fractional peak amplitude m, by comparing inversion ratios z_1 and z_2 at equal time before and after the peak power according to:

$$\epsilon_t(z_t, m) = \frac{ln\, z_t - ln\, z_2}{ln\, z_1 - ln\, z_t} = \frac{w_2(z_t, m)}{w_1(z_t, m)}. \tag{4}$$

From this the ratio of widths w_1 and w_2 at that fractional amplitude can be found [5].

Previous LADAR range estimation algorithms used an inverted paraboloid, a symmetric Gaussian curve, or a polynomial curve to correlate with the incoming measured laser pulse [3]. All of these shapes are symmetric. Siegman and Liu's findings indicate that the pulse is seldom symmetric and can potentially vary based on the processes occurring in the laser cavity. An inappropriate pulse shape model can lead to errors in range estimation algorithms.

2.3 Range Estimation Algorithms

The key objective of any ranging algorithm it to detect the time at which the peak intensity of the reflected return pulse falls incident upon the detector. In order to extract the target range from the 128×128 image, the 19 sample return pulse waveform from each pixel must be processed by a range estimation algorithm. The real target range is identified by the round trip time-of-flight of the peak of the outgoing pulse. The objective of any such algorithm is to correctly determine this peak time from which the target range can be directly calculated. Identification of the peak time is confounded by the presence of noise as well as the pulse shape factors discussed in previous sections. The following sections discuss the prominent estimation algorithms and their relative strengths and weaknesses, especially their susceptibility to errors caused by characteristics of the LADAR return.

2.3.1 Peak Detection. A peak detection algorithm reports the peak time as the time the maximum recorded amplitude occurred. This method is simple, fast and requires no prior knowledge of the pulse shape, but its range accuracy is limited. Without interpolation of the data, this algorithm can only estimate ranges based on one of the 19 recorded slices. With a high signal to noise ratio (SNR), this can result in a potential error of approximately 20 cm if the actual peak time is between recorded slices. However in medium to low SNR conditions where additive noise drives an adjacent slice value higher than the true peak value, the error may be as much as 40 cm. As the pulse shape widens or noise increases, the chance of false maxima increases, which produces erroneous range estimates [3].

2.3.2 Maximum Likelihood Estimators. A maximum likelihood estimator (MLE) is a technique of range estimation using Bayesian analysis tools on probabilities of the random noise in the signal. For a given distribution of noise terms and a predicted shape of the signal, the return pulse can be described by a probability distribution function (PDF) conditioned on the target range. The MLE range estimate is the target range with the highest probability to have generated the return pulse [6]. The MLE improves upon the peak detector in that it can provide range estimates between the recorded slices without introducing interpolation errors. The disadvantage of the MLE is that its accuracy is greatly dependent not only on the quality of the pulse model, but also the quality of the model of the random noise. This method is not ideal due to the varying pulse shape and several sources of noise from different governing distributions present in the LADAR system [4].

2.3.3 Traditional Cross-Correlation. A common method of estimating range from LADAR signals is the signal processing technique of cross-correlation. In this technique the received signal is compared to a shifted version of the original transmitted signal referred to in this thesis as a reference waveform [2]. A mathematical

representation for a discrete cross-correlation of real-valued signals of length N is:

$$\Gamma = \frac{1}{N} \sum_{n=1}^{N} d_n \cdot r_n \tag{5}$$

where d_n are the measured intensity values and r_n are the values of a reference waveform as it is shifted across the frame [8]. The cross-correlation value, Γ, achieves an absolute maximum when the two waveforms are at their best possible alignment. This method assumes that when the two waveforms are aligned, their peaks are also aligned. The corresponding shift can then be related to the time index of the peak of the reference waveform which becomes the estimation of the true peak time of the return waveform.

Cross-correlation is subject to range bias when applied to the waveforms output by the ASC LADAR [3]. The cause of the bias is traced to two main sources. The first source of bias is the truncation of the waveform that occurs when either the leading or trailing edge of the return pulse is cut off by the boundaries of the recorded slices. The second source of bias appears when a shape mismatch exists between the reference waveform and the data waveform.

2.3.3.1 Waveform Truncation Effect on Bias. A peculiar nature of the data collected by the ASC FLASH LADAR was the common appearance of incomplete return waveforms. The problem was most prevalent when the camera was operating in stop mode where the arrival of the return pulse triggered the recording of data. Depending on the preset threshold, the trailing edge of the return pulse was often truncated out of the end of the recorded data.

Waveforms were similarly truncated in SULAR mode when the target was not near the center of the preset range gate. As the target location moved toward the edges of the range gate, an increasing amount of the leading or trailing edge extended outside of the range gate and was not recorded.

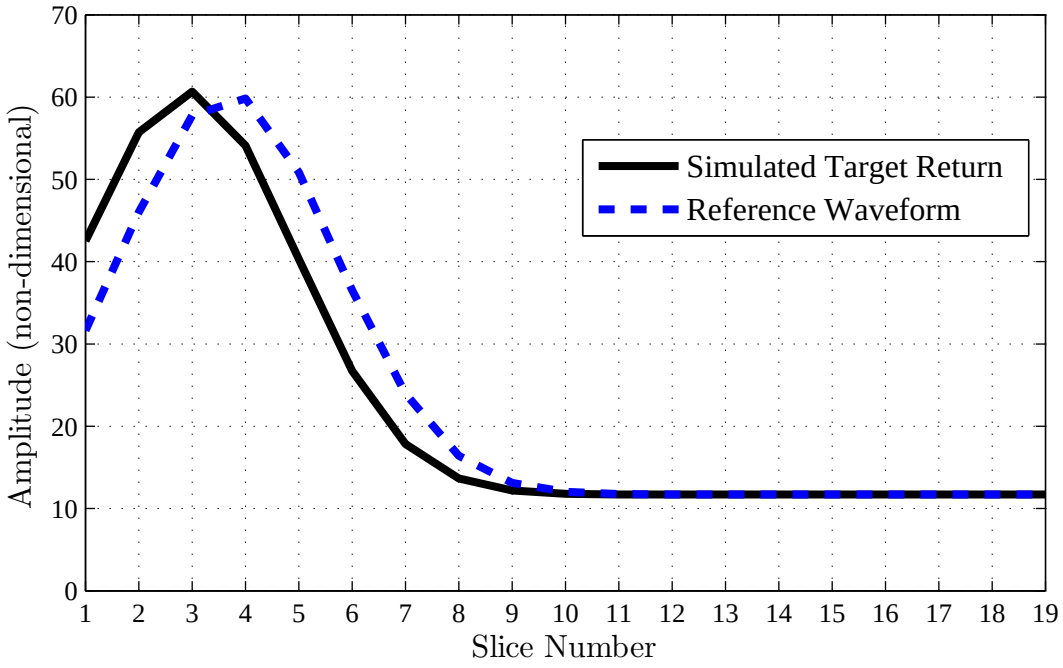

Figure 2: A simulated truncated target return and an associated reference waveform in the position that generated the highest cross-correlation value

Application of the traditional signal processing cross-correlation algorithm to truncated data waveforms produced erroneous range estimations. A trial with simulated data showed the amount of error was directly related to the amount of truncation in the signal waveforms. When truncated waveforms were used, cross-correlation values computed by equation (5) were found to be higher when the reference peak and the data peak were not aligned. Figure 2 shows an example from simulation where the reference at the wrong range resulted in the highest cross-correlation value. In this case, the algorithm returned an estimate with over 30 cm of error.

The source of the error was the asymmetric multiplicative contributions from the non-truncated portion of the waveform. When the reference waveform was shifted further from the edge, the point-by-point product actually increased slightly before dropping off. Since the cross-correlation algorithm returned the location of the reference peak with the highest cross-correlation value, the returned range estimate was

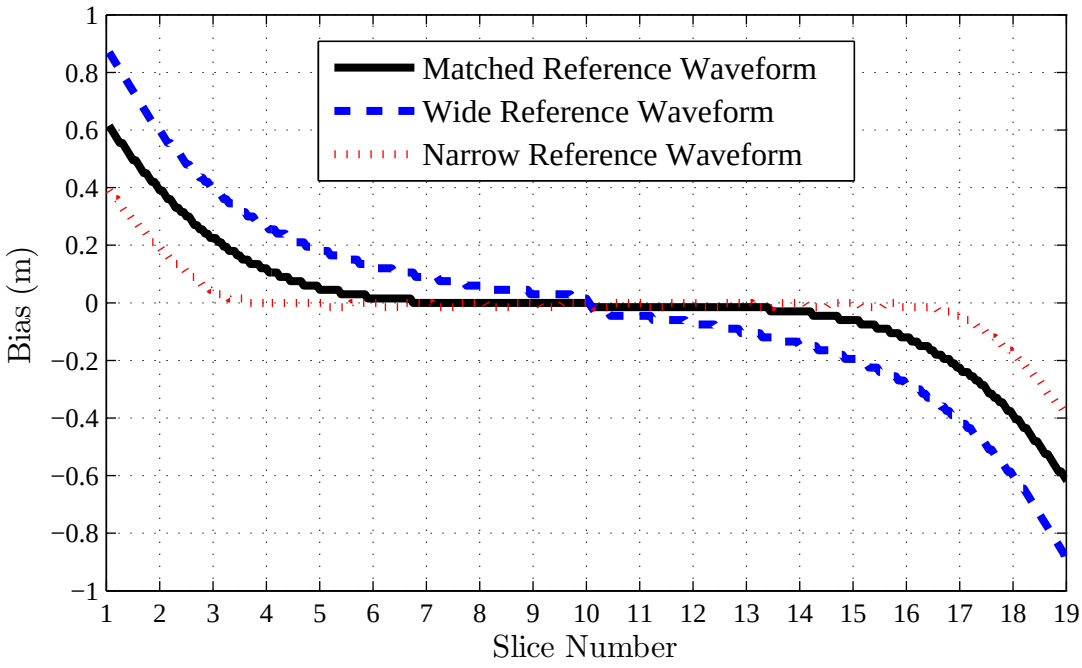

Figure 3: A depiction of how the bias from a cross-correlation range estimation algorithm changes with target location and reference shape

biased toward the center of the range gate. Waveforms with a greater percentage of truncation resulted in a greater bias in the range estimate.

2.3.3.2 Pulse Shape Effect on Bias. Pulse width affected the magnitude of the bias at either end of the range gate. Figure 3 shows patterns of bias produced when a cross-correlation estimation algorithm was applied to a series of simulated target return waveforms swept across the entire range gate. Trials were run with reference waveforms narrower than, equal to, and wider than than the modeled target return waveform. It was observed that a reference pulse wider than the return pulse exacerbated the bias while a narrower reference pulse reduced the bias. A narrow pulse, however ultimately became a peak-finding algorithm and was more susceptible to error caused by under-sampling or noise.

Asymmetric pulse shapes also caused a bias error. For instance, if the target return waveform had a sharply increasing leading edge and a slowly decreasing trailing

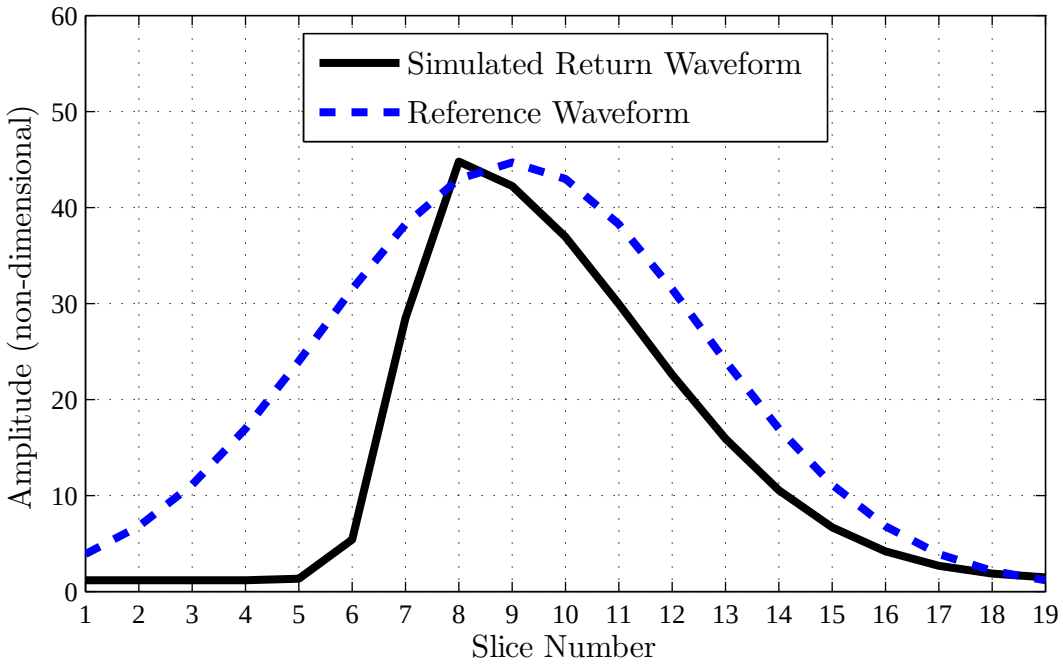

Figure 4: An example from simulation of a 44 cm error caused by a cross-correlation range estimation algorithm using an incorrectly shaped reference waveform

edge, a cross-correlation estimator with a symmetric reference waveform would return a range beyond the actual target range. In all cases, the maximum cross-correlation value would occur when the reference waveform was shifted toward the widest half. Figure 4 shows such a simulated occurrence with a large asymmetry. The data waveform and the reference waveform with the highest cross-correlation value are shown. The range generated by the estimation algorithm was off by 44 cm. The next chapter will discuss a new method for overcoming these susceptibilities and generating a more accurate range estimation from a cross-correlation algorithm.

III. Development and Testing of the NOVAS Range Estimation Algorithm

3.1 Sources of Error in Cross-Correlation Algorithms

Correlation algorithms have several features that make them well suited for range estimation. Unlike a peak-finding estimator, a correlation algorithm can be used to estimate ranges at sub-sample intervals without introducing error from interpolation. Correlation algorithms do not require any prior knowledge of the statistical distribution of the noise, making them generally more applicable than Bayesian likelihood estimators. Correlation algorithms are also straightforward to implement as many computer languages have built-in correlation functions.

However, there are several constraints on the LADAR data which make a cross-correlation algorithm susceptible to error caused by truncation of the received waveform or a shape mismatch between the data waveform and the reference waveform. First, the data waveforms output by the LADAR are entirely positive, and as such so are any suitably matched reference waveforms. Second, the correlation only operates on 19 points across a restricted length of space. Third, any correlation algorithm is a maximizer: the result is obtained by finding the maximum sum of products between the two elements in the correlation. The problem therefore becomes setting boundaries on the reference waveforms to find the highest meaningful maximum. For the purpose of range estimation, a meaningful maximum occurs when the highest correlation value generated between the reference waveform and the data waveform indicates that the peaks of the two waveforms are aligned in range. For this discussion, a false maximum is defined as a maximum correlation value that occurs when the peaks of the data waveform and the reference waveform are not aligned.

Figure 5 demonstrates that for a given symmetric data waveform without significant truncation, wider or higher amplitude reference waveforms will result in a higher correlation value simply because there are a greater sum of products between positive numbers. In this case, range algorithms based on these reference waveforms will produce the same range estimate—all the peaks are perfectly aligned. Although

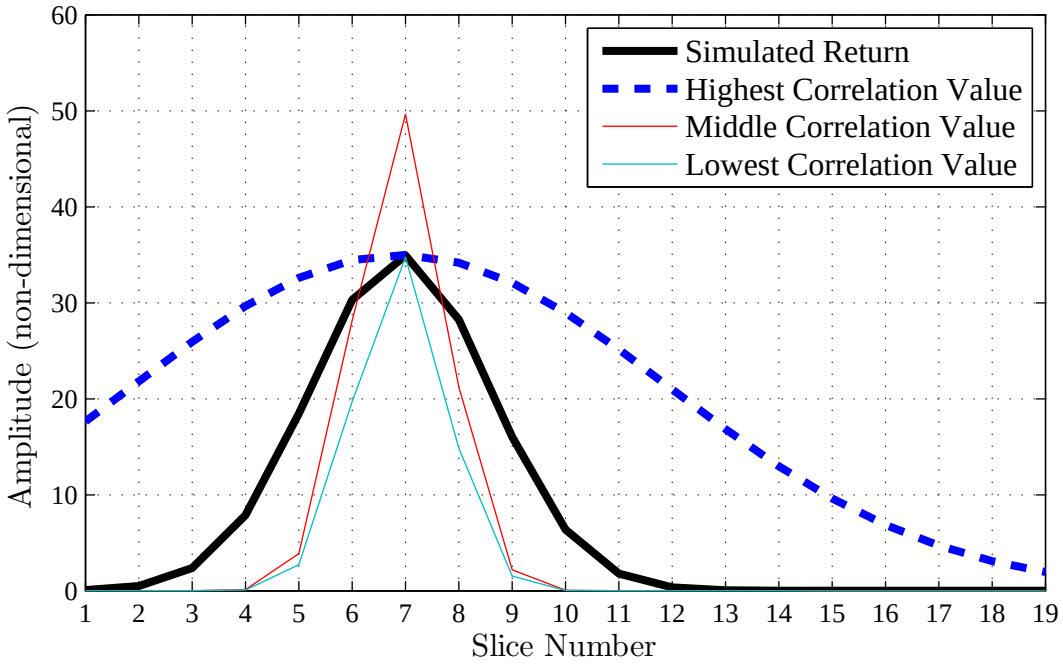

Figure 5: A demonstration that a cross-correlation algorithm will have a higher correlation value with wider or higher amplitude reference pulses

the waveforms are not identical, the output is a meaningful maximum. This is the condition for which traditional cross-correlation algorithms are designed to operate. However, as seen in Chapter 2, truncation and shape mismatch exist and induce error into such algorithms. These errors also result from the positive nature of the waveforms creating undesirable contributions to the total correlation value when an asymmetry exists in the data waveform.

3.2 Normalized Variable Shape Correlation

The solution proposed in this thesis is the Normalized Variable Shape (NOVAS) correlation algorithm which attempts to correct the errors discussed above while preserving the advantages of a correlation algorithm. The key functionality is two-fold. First, the proposed correlator not only scans across the range gate, but also scans across an available set of possible pulse shapes. The correlator also calculates the

normalized Pearson's product-moment correlation coefficient. These two elements working in concert work to provide a versatile range estimation algorithm.

3.2.1 Pearson's Product-Moment Coefficient. Use of the Pearson's product-moment correlation coefficient eliminates truncation and pulse shape effects from a correlation estimator. The coefficient uses the mean amplitude of the data and reference waveforms, $\bar{d}$ and $\bar{r}$ respectively, as well as the standard deviation of the amplitude values of each waveform, σ_d and σ_r, to normalize the value. It is calculated by:

$$\rho = \frac{1}{N} \sum_{n=1}^{N} \frac{(d_n - \bar{d})(r_n - \bar{r})}{\sigma_d \sigma_r}, \tag{6}$$

and takes on values between one and negative one indicating the linear association between the two waveforms [12].

The first effect of this calculation is to create two zero-mean waveforms with positive and negative values. This allows portions of the waveforms that are poorly aligned to subtract from the correlation value. The defining features of these waveforms become their origins when they cross the zero-axis. Two waveforms of matched shape will produce an absolute maximum or minimum correlation value when they have equal numbers of origins, and all origins are aligned. Since the waveforms are of matched shape, alignment of the origins equals alignments of their peaks and an accurate range estimation can be produced regardless of truncation.

When discussing waveforms of matched shape, the amplitude of the waveforms is irrelevant. In fact the only feature that must match is the pulse widths, which are defined for this thesis as the distance from the position of peak amplitude to one standard deviation of amplitude to each side.

If two Gaussian waveforms have the same pulse-width, their mean values will occur at the same distance from the pulse center. Thus two zero-mean waveforms with equal pulse-widths can be aligned such that their origins are superimposed regardless

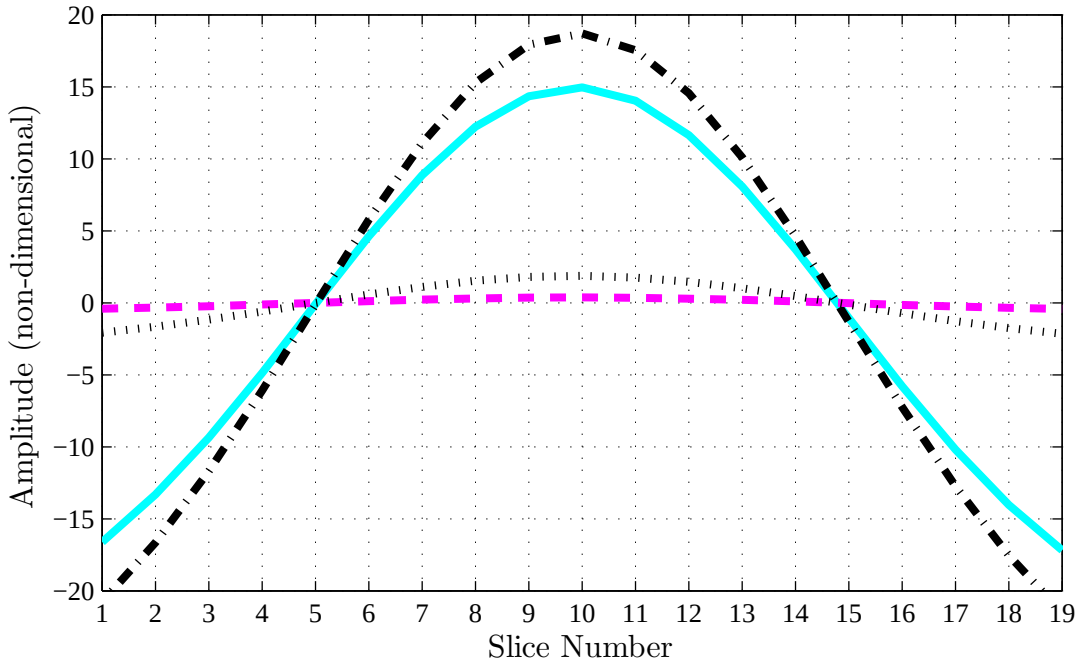

Figure 6: An example of zero-mean waveforms with various amplitudes but equal pulse widths that have aligned peaks when their origins are aligned

of amplitude, as demonstrated in Figure 6. If these pulses are then normalized, that is divided by their respective standard deviations, they they become the same pulse.

Among a set of zero-mean, normalized waveforms, the absolute maximum correlation value will occur when the two waveforms in the correlation are the exact same shape and are perfectly aligned at every point. Only in this case will every product included in the summation shown in equation (6) be a positive number. If one waveform is shifted or is a slightly different shape, negative numbers appear in the summation resulting in a reduced correlation value.

3.2.2 NOVAS: A Range and Pulse Shape Correlation Algorithm. A range estimation correlation algorithm using Pearson's correlation coefficient and an appropriate set of reference waveforms eliminates errors caused by waveform truncation and pulse shape effects. The Pearson's coefficient simultaneously characterizes correlation of the position and shape of two waveforms and achieves an absolute maximum value when the reference pulse is aligned with the return waveform and has the same pulse

shape. This allows the NOVAS algorithm to scan a multitude of pulse shapes across the range gate achieving a separate Pearson's coefficient for each distinct reference pulse shape at each range location. The single highest coefficient value simultaneously indicates the shape and position of maximum correlation, returning an unbiased range estimate regardless of target position or return pulse shape.

3.2.3 NOVAS Model of LADAR Pulse. To implement the pulse shape estimator, the actual target return pulse shape was modeled as a continuous piecewise Gaussian curve. The model consisted of the left and right side of two different Gaussian curves joined at their peak. Each half was non-normalized but of equal amplitude such that the resulting joined curve was continuous and smooth. Each half of the pulse could be assigned a separate pulse-width such that asymmetric pulses could easily be modeled by:

$$A(t) = e^{\frac{-(t-2R_{target}/c)^2}{2\sigma^2}} \begin{cases} \sigma = \sigma_{left} & \text{if } t \leq 2R_{target}/c \\ \sigma = \sigma_{right} & \text{if } t > 2R_{target}/c. \end{cases} \tag{7}$$

In this equation, $A(t)$ is the amplitude of the pulse, R_{target} is the actual target range, and t is the time elapsed since the laser pulse was transmitted.

The model of the data waveform recorded by the ASC FLASH LADAR was of the same form, however it was a set of 19 discrete samples from this continuous waveform given by:

$$D_k = e^{\frac{-(t_k-2R_{target}/c)^2}{2\sigma^2}} \begin{cases} \sigma = \sigma_{left} & \text{if } t_k \leq 2R_{target}/c \\ \sigma = \sigma_{right} & \text{if } t_k > 2R_{target}/c. \end{cases} \quad \text{for } k \in 1, 2, \ldots, 19. \tag{8}$$

The waveform was recorded in the form of digital counts measured across the integration time of the detector. In this equation, R_{target}, σ_{left}, and σ_{right} are the unknowns to be estimated. This model allows R_{target} to be any real number between the min-

imum and maximum ranges on the range gate, thus not limiting range accuracy to the 19 data points actually measured.

3.2.4 Correlation Method. The correlation method used was based on a slide–and–multiply type of correlation, however it also took advantage of efficient matrix operations in `Matlab`® . The actual correlation was performed by multiplying the 1×19 data waveform row vector by a $19 \times M$ reference matrix, where M is the number of unique parameter combinations. Each column of the reference matrix was a single 19×1 reference waveform defined by a range, a left pulse width, and a right pulse width.

The reference waveform was constructed in the same manner as the data waveform in equation (8), however specific values for R_{target}, σ_{left}, and σ_{right} were chosen. Each choice of these variables created a separate reference waveform vector available to correlate with the data vector. Again, because R_{target} could be chosen from any real number in the range gate, the range estimation accuracy of this algorithm was not limited by the discrete number of data points. Furthermore, the correlation only occurred on the 19 measured data points and required no interpolation of the data waveform. These features allowed sub-sample accuracy to be achieved without any error injected by interpolation.

Prior to the vector-matrix multiplication, the data waveform and each column of the reference waveform were normalized in the manner of the Pearson's coefficient. The result of the entire operation was a $1 \times M$ row vector of Pearson's coefficients given by:

$$\rho_m = \frac{1}{19} \sum_{k=1}^{19} \frac{(d_k - \bar{d})(r_{k,m} - \bar{r}_m)}{\sigma_d \sigma_{r_m}} \quad , m \in 1, 2, \ldots, M. \tag{9}$$

The reference waveform with the highest correlation value was indicated by the column index of the highest valued correlation coefficient. With careful index accounting, the location of the peak and the left and right pulse widths of the selected reference wave-

form could be extracted, giving the target range estimate and a piecewise Gaussian estimate of the target return pulse shape.

3.2.5 Building the Reference Matrix. The number of reference waveforms, M, in the reference matrix depended upon the range accuracy or flexibility desired from the algorithm. An increased number of waveforms allowed range estimation to a smaller sub-interval, applicability to a more diverse set of pulse shapes, or both.

3.2.5.1 Range Sub-Intervals. The range accuracy of the algorithm was in part dependent on the number of sub-sample ranges, L, chosen between each of the 19 measured data points. A smaller interval meant more reference waveforms vectors, a larger reference matrix, and longer computation times. This was assuaged somewhat by assuming that the actual peak of the return waveform occurred within a certain distance of the peak measured data point, and thus only a portion of the entire range gate must be searched for R_{target}. For this thesis, the reference waveforms swept across five slices of range centered upon the slice with the maximum amplitude return. This resulted in $(4L + 1)$ individual range increments.

3.2.5.2 Pulse-Widths. The selection of pulse-widths available to the algorithm also affected the reference matrix size, computation cost, and to some extent the range accuracy. The reference matrix was created by choosing a left and right pulse-width from a set of N available pulse widths. This allowed a collection of $(N^2 - N)$ distinct symmetric and asymmetric pulse shapes that could be used in the correlation. Each distinct waveform was then centered at each sub-sample range, giving the total number of waveforms used in the correlation as $M = (4L+1)(N^2-N)$. Thus the computational cost of the algorithm was much more sensitive to the number of available pulse-widths than to the sub-sample range interval.

The importance of this lay in the ability to provide the right pulse-widths to achieve the most accurate range estimation. If the actual width of the return pulse was between available widths used to build the reference matrix, an inaccurate range

estimation occurred. Following sections will quantify this error and describe characteristics of the ASC FLASH LADAR from which a suitable set of pulse-widths could be determined.

3.3 Performance of NOVAS in Simulation

This section discusses various aspects of performance of NOVAS compared to the traditional cross-correlation algorithm. The simulation was run in a series of trials using simulated LADAR returns generated in the same manner as the reference waveforms used in the NOVAS algorithm. The simulation was generated with the condition that the electronic oscillator was operating at the nominal frequency of 400 MHz, with the distance between slices representing 0.35 m or 1.2 ns. The first trial demonstrated how the new algorithm eliminated the errors of previous algorithms in ideal conditions. Second, it will be shown how a non-ideal reference matrix affected the range accuracy of the algorithm.

3.3.1 Ideal Conditions. This section compares the baseline performance of NOVAS against that of a traditional cross-correlation. Simulated noise-free waveforms were used, and the reference matrix used in the new algorithm was ideal—it contained the the precise pulse-widths and target ranges presented in the simulated data waveforms. The cross-correlation algorithm was also given the precise waveform and target ranges for each trial to ensure a level playing field. Technically, this simulation compared one realization of the proposed algorithm against numerous realizations of a cross-correlation, since no single version of the latter had the flexibility to be applied adequately to the range of scenarios presented.

The data waveforms used were not necessarily representative of the ASC FLASH LADAR waveform, but were generalized to show the value and adaptability of the normalized correlator. The effects demonstrated in the generalized simulation certainly are mirrored in the specific case of the ASC FLASH LADAR.

Figures 7 through 12 illustrate that between NOVAS and cross-correlation, only the normalized correlation algorithm will choose the correct shape and range when given a choice of a variety of each. Each figure has four sub-figures. Sub-figure *a*) shows a real-world, continuous time representation of the return waveform and the particular reference waveform used. The data and reference waveform were deliberately set at different amplitudes as the amplitude of the return waveform would always be unknown. Sub-figure *b*) shows the 19 sample discrete representation of the same waveforms that were output by the LADAR system and used by both correlation techniques. None of the waveforms displayed have been normalized for better visualization. Sub-figure *c*) shows a bar-graph representation of the normalized data and reference waveform amplitudes as well as the point-by-point product of each of the 19 amplitudes. The Pearson's product-moment correlation coefficient, ρ, is indicated as well. Sub-figure *d*) shows a bar-graph representation of the non-normalized waveforms and the cross-correlation value, Γ. The point-by-point products in sub-figure *d*) have been reduced by a factor of 10 for ease of display.

Figure 7 shows a simple case: a symmetric Gaussian pulse centered in the range gate with no significant truncation. The reference pulse was matched in shape and location giving a Pearson's coefficient, ρ, equal to one. Note that in Figure 7 c) the values of the normalized waveforms ranged through positive and negative numbers, yet their product was always non-negative. This was always true when the reference location and pulse-width were perfectly matched to the data waveform.

In Figure 8, the reference pulse was in the correct location, but it was wider than the data pulse. It is important to note that in this situation, ρ decreased while the output of the cross-correlation Γ increased. Although the cross-correlation produced the right range estimate in this case, it became clear that it was affected by the pulse shape.

Figure 9 also shows a symmetric return pulse, however it was significantly truncated at the end of the range gate. NOVAS again found a maximum correlation value

equal to one when the reference pulse matched in shape and location. Figure 10 shows that a correctly shaped pulse achieved a maximum cross-correlation value when it was biased toward the middle of the range gate, a problem exacerbated by a mismatched reference pulse shape as shown in Figure 11. NOVAS was not susceptible to these errors.

Figure 12 shows an asymmetric pulse without significant truncation. Although it was given a well matched reference waveform in both shape and range, the cross-correlator did not produce the maximum value for Γ in these conditions. Rather, Γ increased with a mismatched shape, as seen in Figure 13, and increased further with an incorrect range, as seen in Figure 14. In these cases, the reference waveform was of similar width but was perfectly symmetric. Figure 14 shows that in the cases of asymmetric data waveforms, Γ was highest when the reference waveform was biased toward the wider portion of the asymmetric data waveform. At the same time ρ remained slightly, but definitively, less than one in all cases that either the pulse shape or range were not matched precisely.

3.3.2 Non-Ideal Conditions. This section presents analysis of the NOVAS algorithm in non-ideal conditions. Presented are characterizations of the error introduced when the set of available reference waveforms under-samples in range or pulse-width.

3.3.2.1 Under-Sampled Range. To characterize under-sampling in range, an asymmetric simulated data waveform was swept across the range gate at range intervals of 0.001 m or 300 steps per slice of the recorded waveform, generating 5,401 data waveforms. The left pulse width was 3 ns and the right pulse width was 8 ns, similar to those used in Figure 14. The pulse was characteristic of NOVAS estimates achieved using LADAR waveforms previously gathered by AFIT personnel. NOVAS was used to estimate a range for each of the 5,401 data waveforms. The set of reference waveforms included the actual waveform used as the simulated LADAR return, but also allowed the pulse width of each side to vary by up to 1 ns at intervals

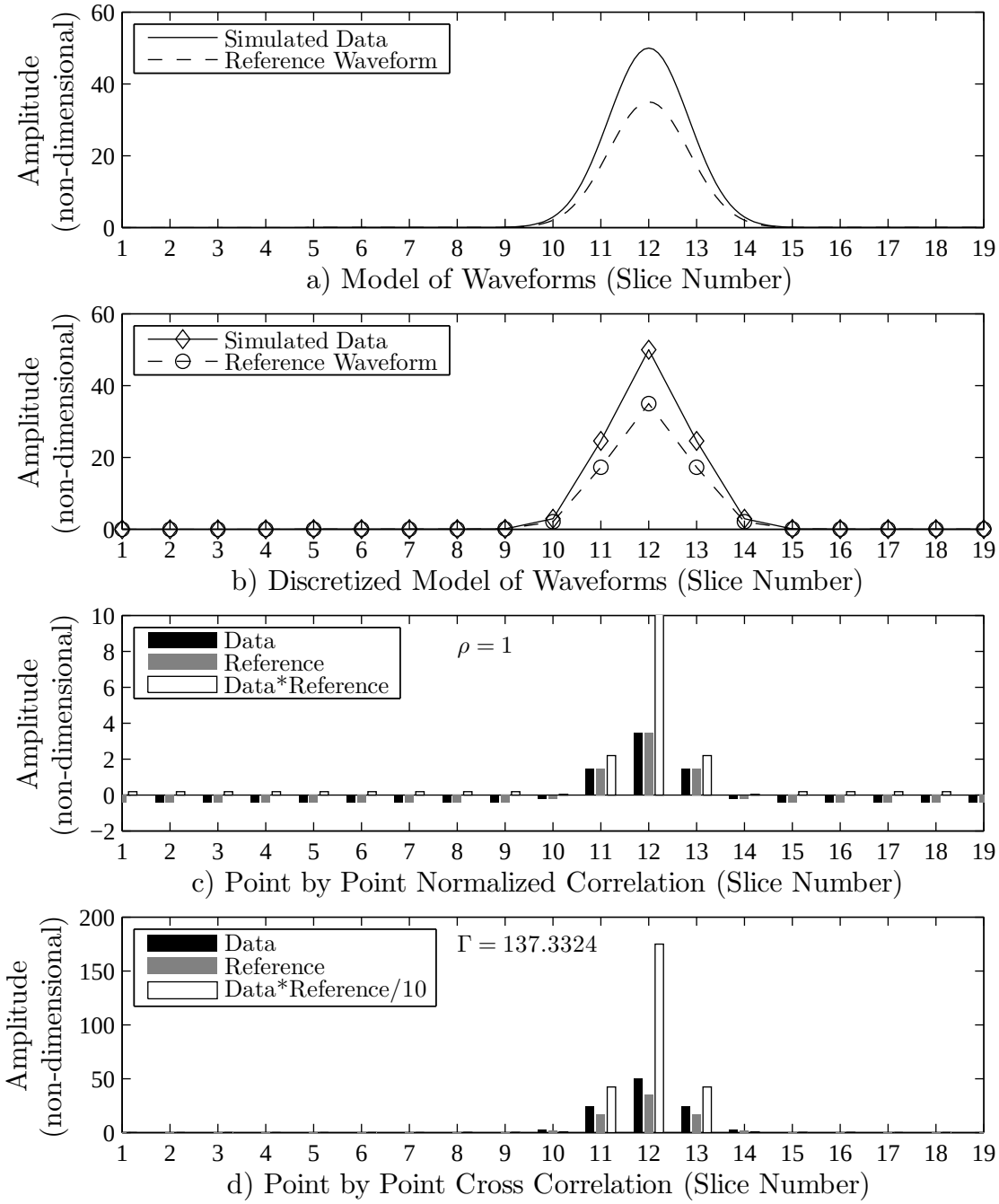

Figure 7: A comparison of normalized and non-normalized correlation when the shape and peak of the reference waveform matched the simulated data waveform; note $\rho = 1$

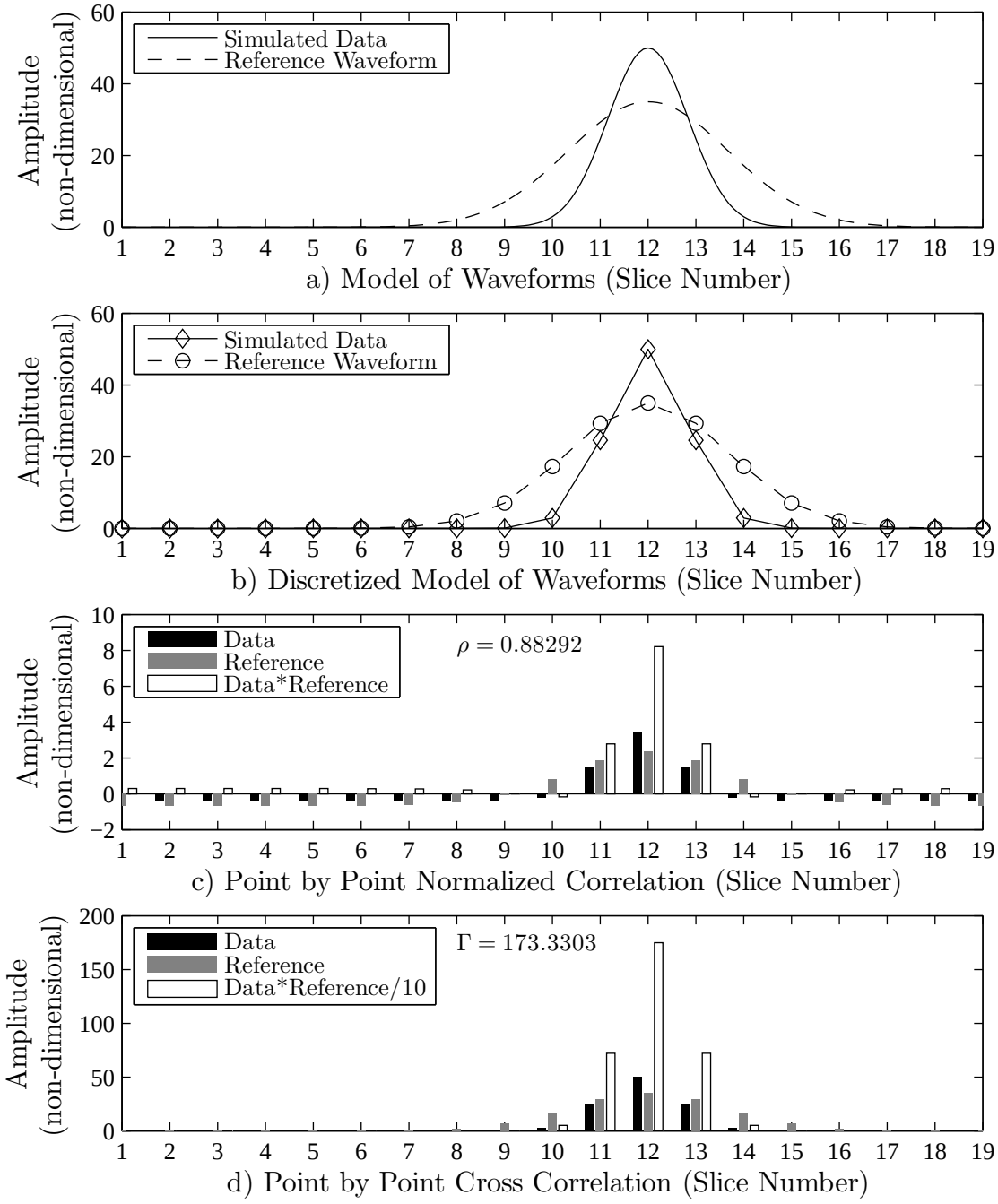

Figure 8: A comparison of normalized and non-normalized correlation when the reference waveform was wider than the simulated data waveform but the peaks were aligned; note $\rho < 1$ but Γ was relatively large

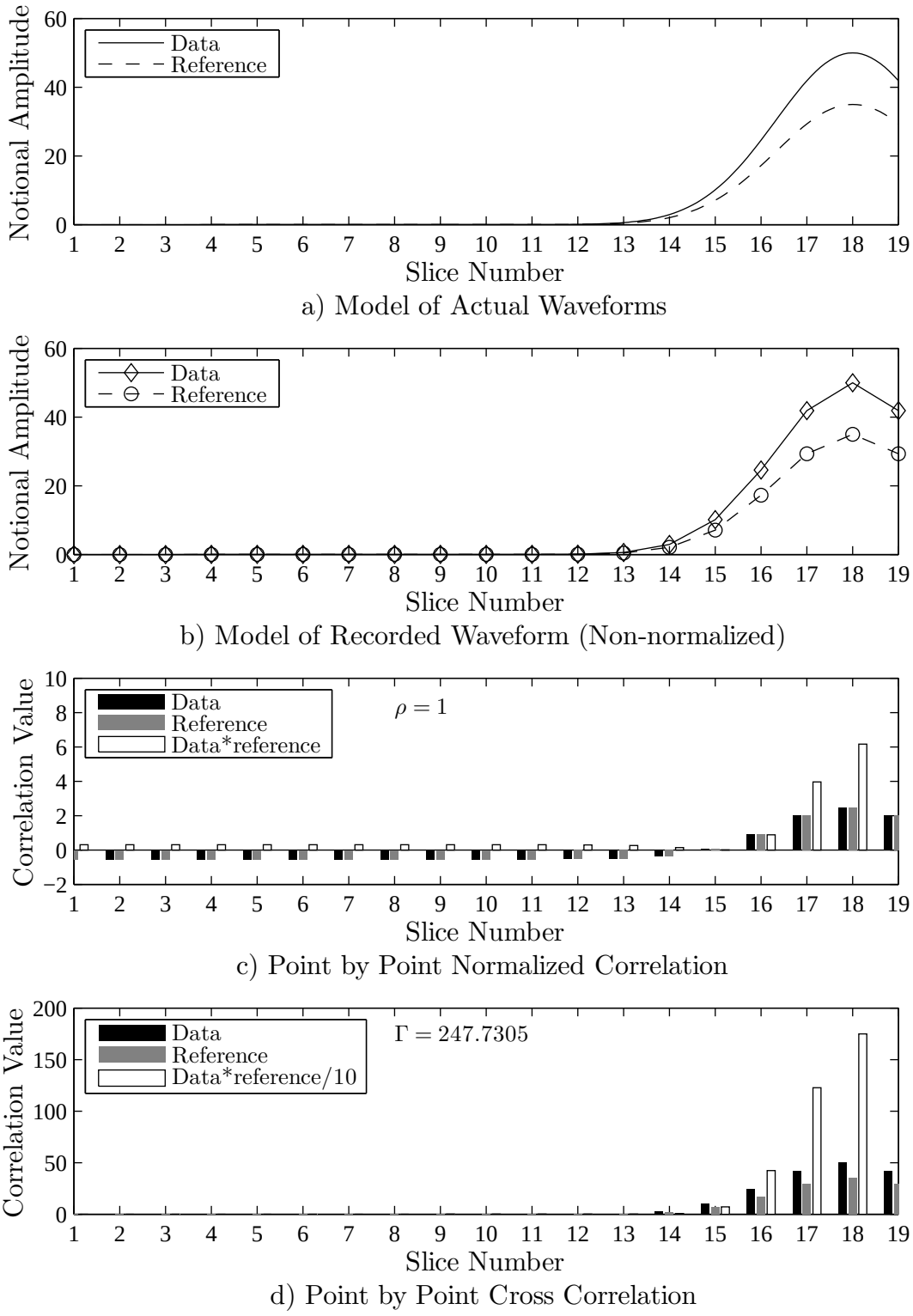

Figure 9: A comparison of normalized and non-normalized correlation when the shape and peak of the reference waveform matched the truncated simulated data waveform; note $\rho = 1$

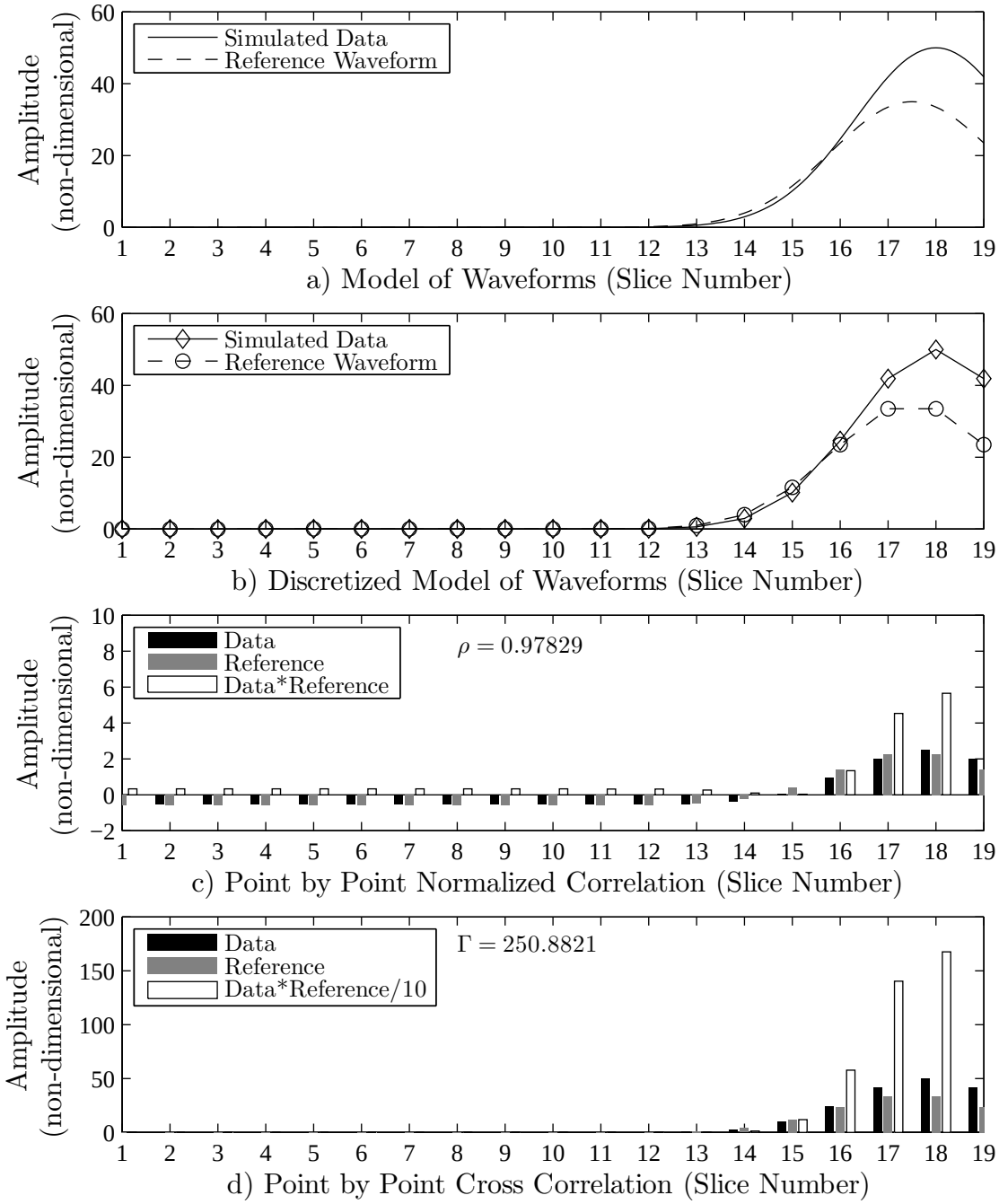

Figure 10: A comparison of normalized and non-normalized correlation when the shape of the reference waveform matched the truncated simulated data waveform but the peak was biased left; note $\rho < 1$ but Γ was greater than that of Figure 9

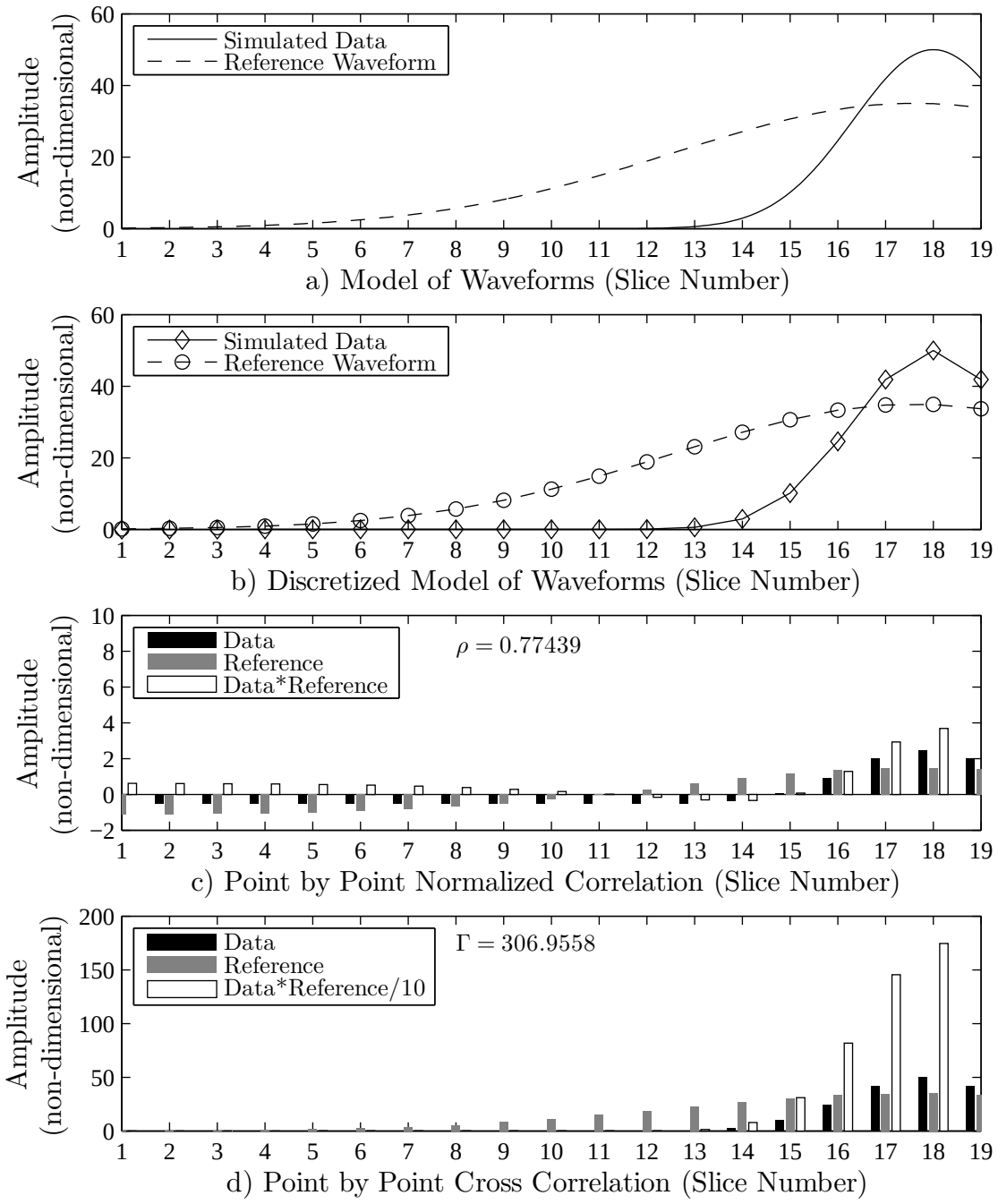

Figure 11: A comparison of normalized and non-normalized correlation when the reference did not match the simulated data peak or shape; note ρ decreased but Γ was greater than in Figures 9 or 10

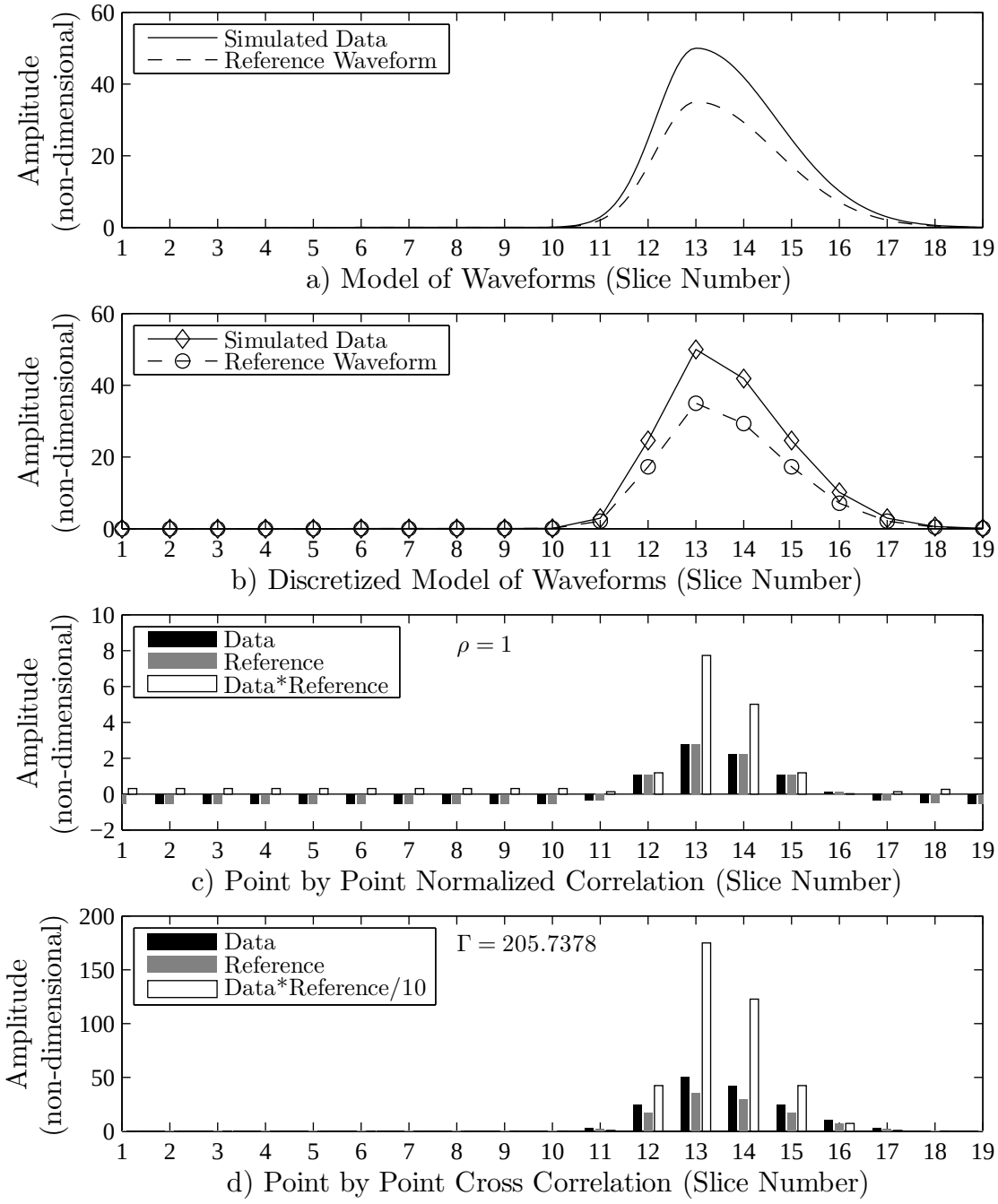

Figure 12: A comparison of normalized and non-normalized correlation when the asymmetric simulated data and reference waveforms matched; note $\rho = 1$

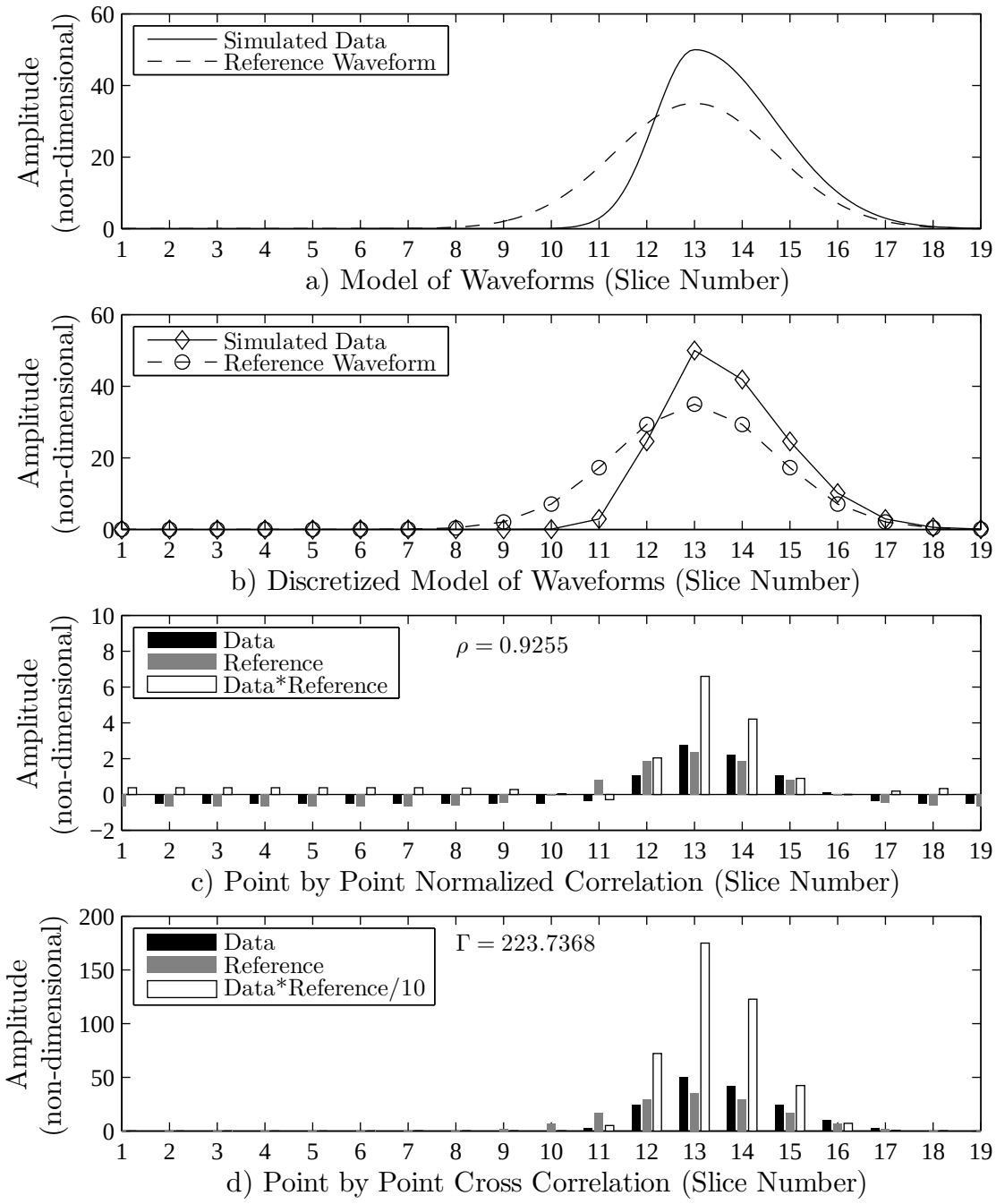

Figure 13: A comparison of normalized and non-normalized correlation when the reference waveform had a similar width and the same peak as the simulated data waveform

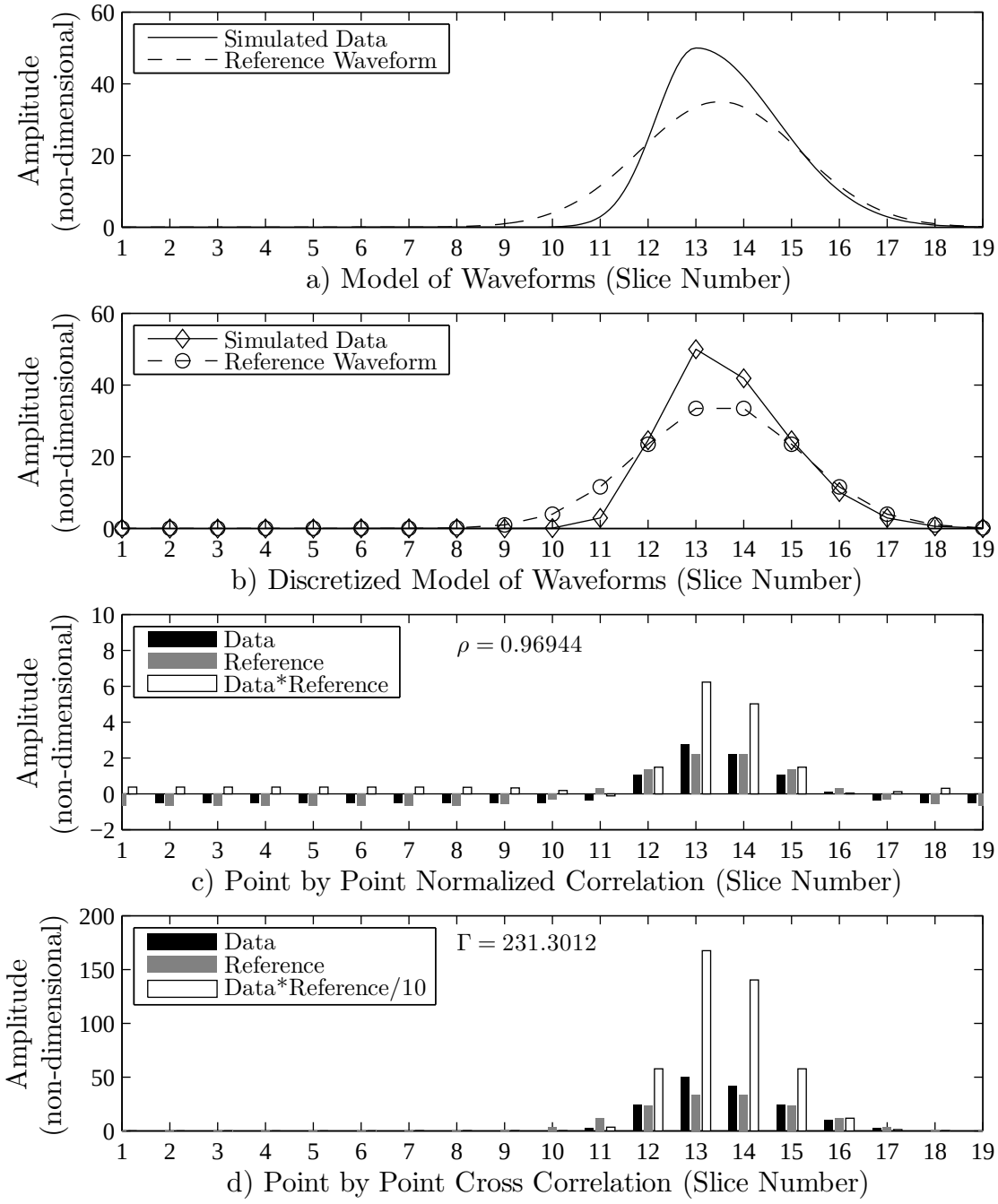

Figure 14: A comparison of normalized and non-normalized correlation when the reference waveform had a similar width as the simulated data waveform but shifted peak; note the higher value for Γ than in Figure 13

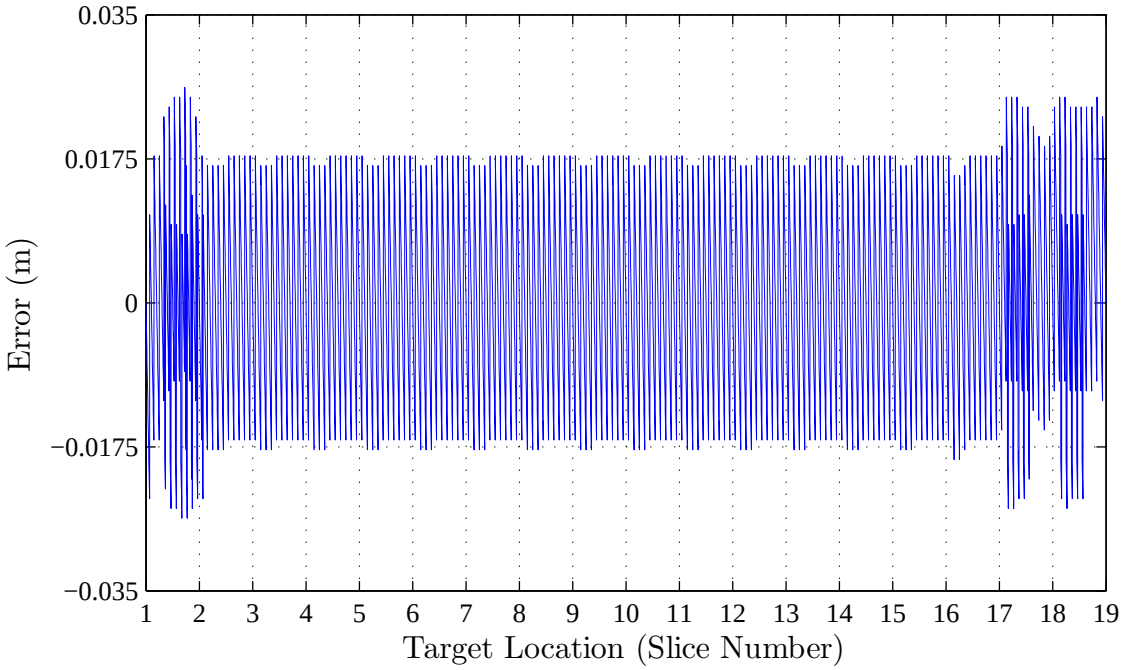

Figure 15: An example of range error versus target location when the NOVAS reference matrix is under-sampled in range

of 0.1 ns. However, the ranges available to the reference waveforms were limited to intervals of 0.0035 m, or 100 steps per slice of the recorded waveform. Thus only every third data waveform was allowed to be ranged exactly.

Figure 15 shows the resulting error in each of the 5,401 range estimates. Aside from the extreme ends, the maximum error is exactly half the sample spacing of the reference waveforms. At the ends, the estimate suffers some error due to truncation of the rising pulse. The root cause is an error in estimating the left and right pulse-widths as the number of data points become fewer and fewer at the respective ends. The wider right half suffers sooner than the left half, however this error is significantly smaller than the bias suffered at the endpoints of a non-normalized correlator as seen in previous sections.

3.3.2.2 Under Sampled Pulse-widths. As seen in the endpoints of Figure 15, error in pulse-width estimation translates to error in range estimation. To characterize this error, the same data waveform was swept across the range gate

at 50 steps per slice while the reference ranges were over sampled at 100 steps per frame. This allowed exact ranging but also allowed error in increments of 0.0035 m. In this case, the available reference pulse-widths did not include 3 ns or 8 ns, forcing a minimum pulse-width estimation error based upon the spacing used to build the reference matrices.

For the first trial, NOVAS had to choose pulse-widths from the sets:

$$\sigma_{left} = [1.5\ 2.5\ 3.5\ 4.5] \text{ ns}$$

and

$$\sigma_{right} = [6.5\ 7.5\ 8.5\ 9.5] \text{ ns}$$

.

The error from this trial is shown in Figure 16. The estimator never achieved the correct range estimate while the pulse width estimates bounced between the two closest pulse-widths available. Only at the end points did the estimator choose from the widest of the available pulse-width, significantly affecting the range estimate in one location. For a minimum pulse-width error of 0.5 ns The mean absolute error was 8.46 cm.

For the second trial, shown in Figure 17, the available pulse-widths were the sets:

$$\sigma_{left} = [2.5\ 2.75\ 3.25\ 3.5] \text{ ns}$$

and

$$\sigma_{right} = [7.5\ 7.75\ 8.25\ 8.5] \text{ ns}$$

.

The minimum pulse-width error was 0.25 ns, but the estimator still never achieved the correct range estimate. The mean absolute error was 4.21 cm. Additionally, a greater percentage of the error was biased toward the wider half of the pulse, similar to the asymmetric mis-match seen in Figure 4.

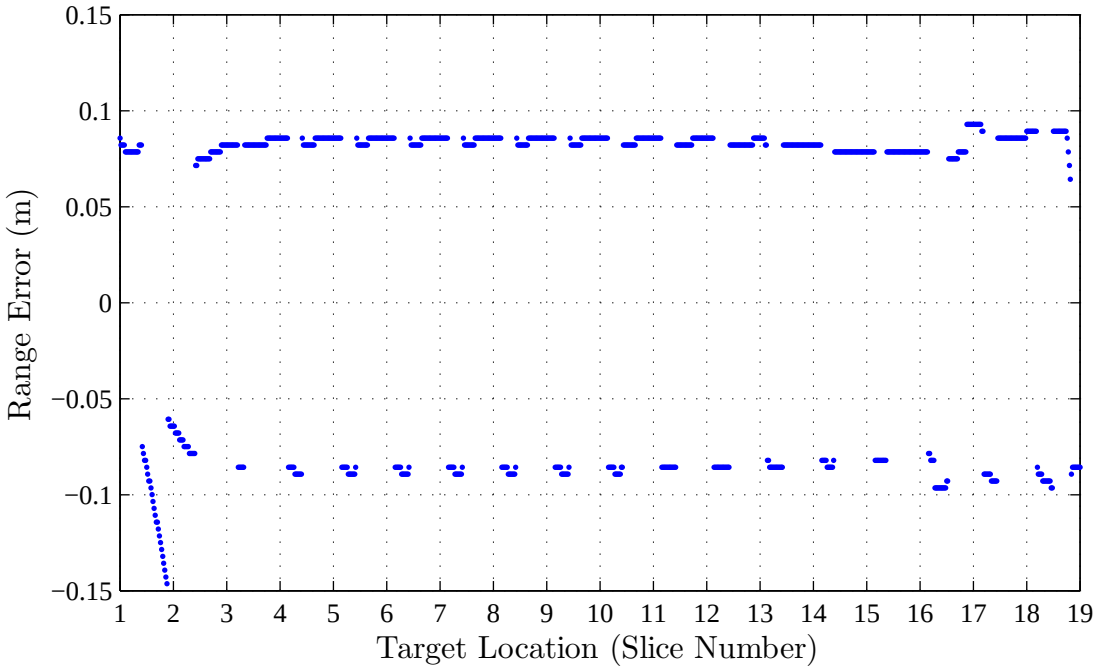

Figure 16: The error in NOVAS range estimations due to a minimum 0.5 ns pulse-width error

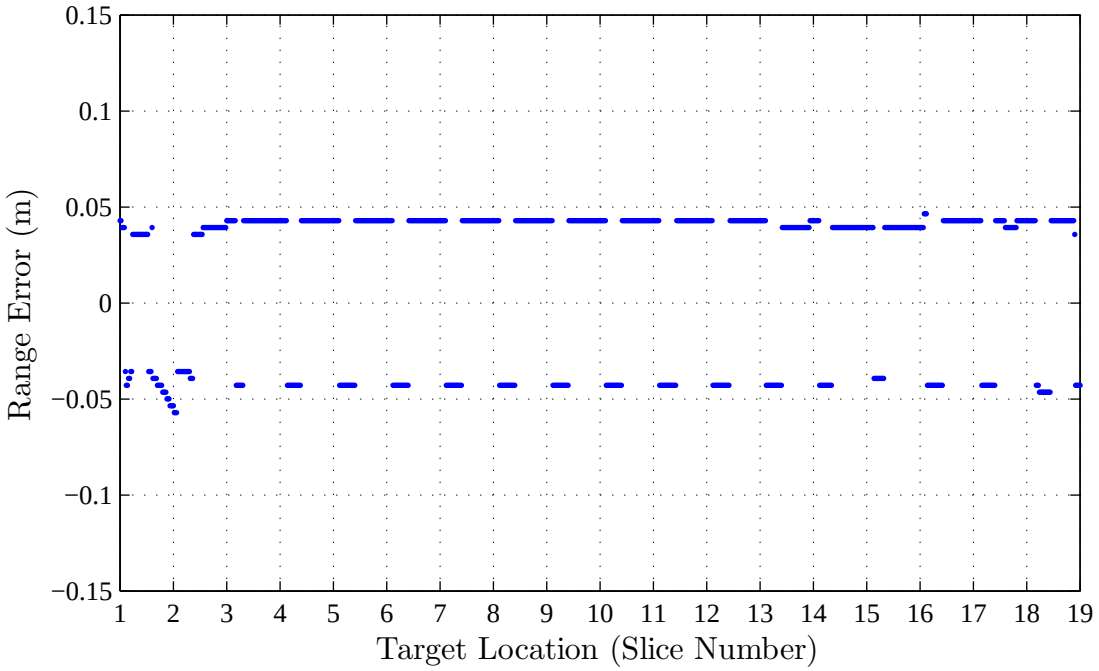

Figure 17: The error in NOVAS range estimations due to a minimum 0.25 ns pulse-width error

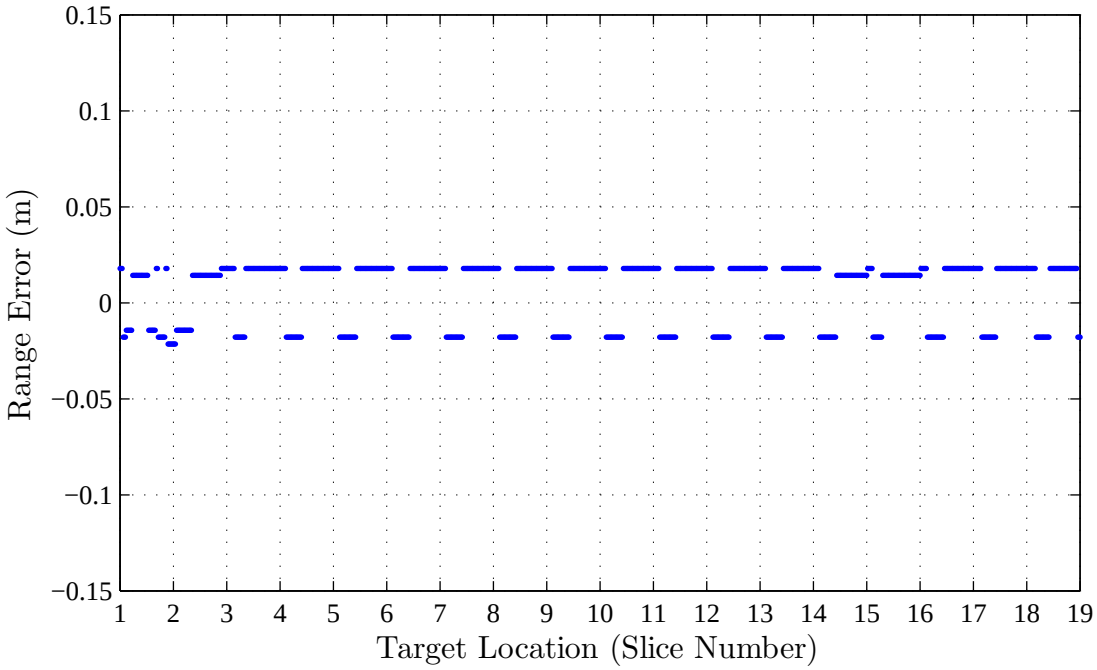

Figure 18: The error in NOVAS range estimations due to a minimum 0.1 ns pulse-width error

For the third trial, shown in Figure 18, the available pulse-widths were the sets:

$$\sigma_{left} = [2.8\ 2.9\ 3.1\ 3.2] \text{ ns}$$

and

$$\sigma_{right} = [7.8\ 7.9\ 8.1\ 8.2] \text{ ns}$$

.

The minimum pulse-width error was 0.1 ns, and the mean absolute error was reduced to 1.74 cm, heavily weighted toward the wider half of the pulse.

For the fourth trial, shown in Figure 19, the available left and right pulse-widths were the sets:

$$\sigma_{left} = [2.9\ 2.95\ 3.05\ 3.1] \text{ ns}$$

and

$$\sigma_{right} = [7.9\ 7.95\ 8.05\ 8.1] \text{ ns}$$

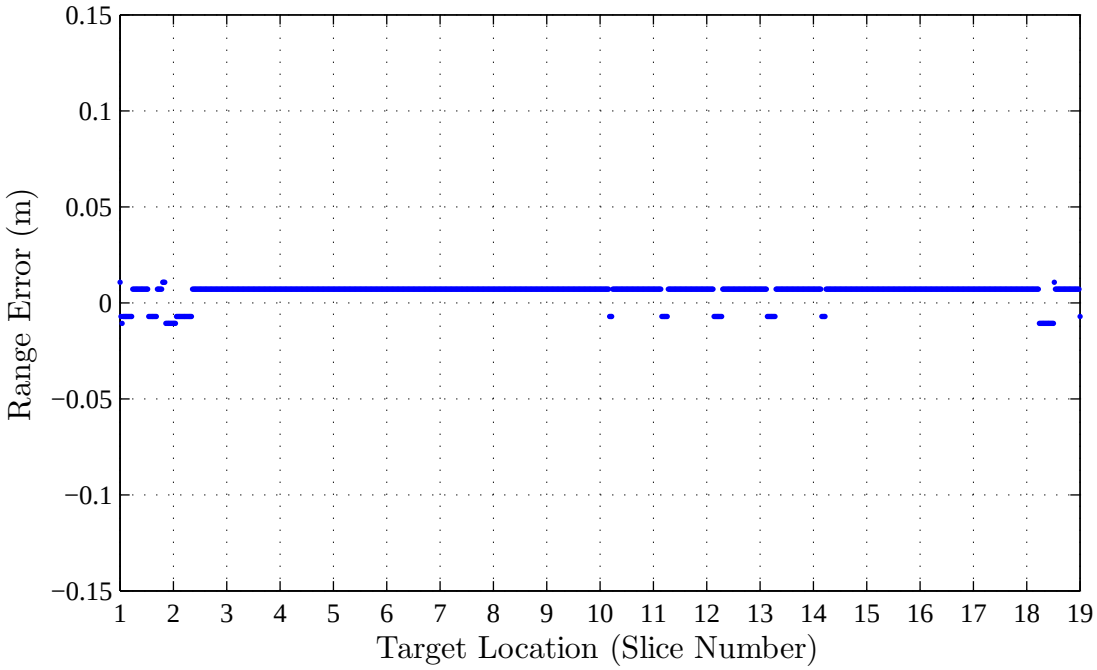

Figure 19: The error in NOVAS range estimations due to a minimum 0.05 ns pulse-width error

.

The minimum pulse-width error was 0.05 ns, and the mean absolute error was further reduced to 0.73 cm, almost entirely weighted toward the wider half of the pulse.

While the second through fourth trials did not include the potential for maximum error at the endpoints, trial five shows that including such errors would not affect the results significantly. For this trial, the available left and right pulse widths were from the sets:

$$\sigma_{left} = [1.5\ 2.5\ 2.95\ 3.05\ 3.5\ 4.5] \text{ ns}$$

and

$$\sigma_{right} = [6.5\ 7.5\ 7.95\ 8.05\ 8.5\ 9.5] \text{ ns}$$

,

allowing a maximum error of 1.5 ns. The results are shown in Figure 20. The mean absolute error only increased to 0.75 cm and the maximum error was only 1.79 cm.

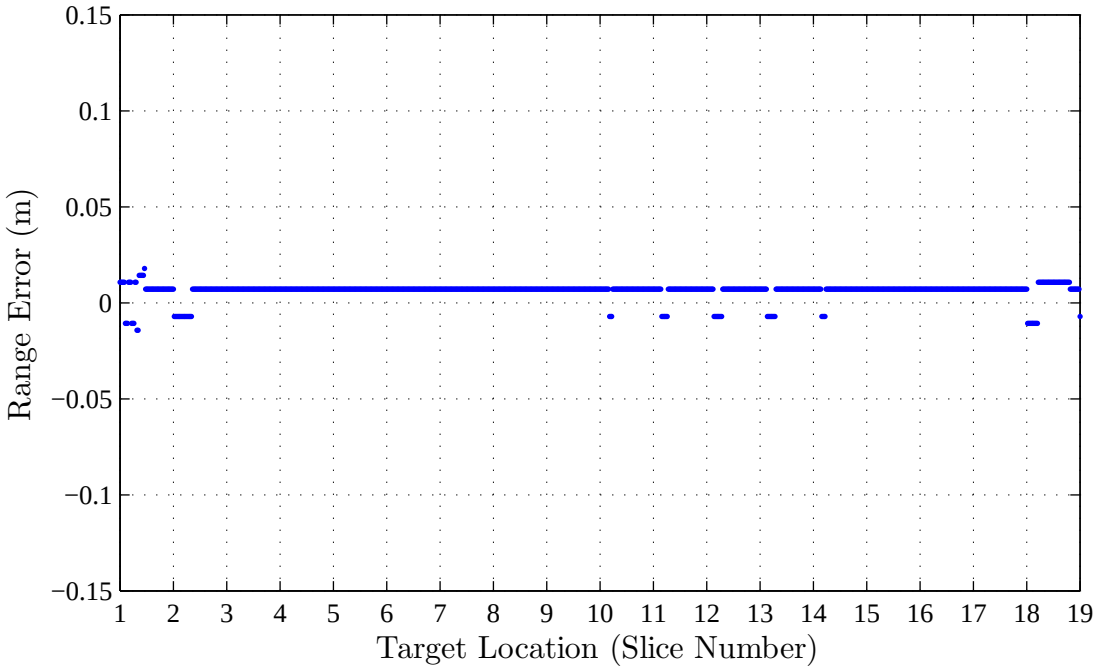

Figure 20: The error in NOVAS range estimations due to a minimum 0.05 ns pulse-width error, but allowing a maximum 1.5 ns error.

This is significantly less than in the first trial which also allowed the same maximum error. Thus, as long as one half of the pulse could be accurately estimated, the error from the truncated half could be greatly minimized.

IV. Ground Test Execution and Results

4.1 Test Execution

In preparation for future flight test at the United States Air Force Test Pilot School (TPS) the ASC LADAR underwent two ground tests under the test management project (TMP) name *Walker Ranger*. For both ground tests the LADAR system was mounted in the RASCAL pod in the same manner as it would be for flight test. The LADAR system was configured in the pod to image targets at a 60 degree angle aft and downward of the pod through a polycarbonate viewing window. To accomplish this, the LADAR optics and detector were mounted such that the field of view (FOV) was oriented pointing aft parallel to the longitudinal axis of the pod. An image re-direction mirror was mounted in the pod just aft of the optics to redirect the FOV in the desired direction. Figure 21 shows this configuration which was designed for use in flight to be able to image an aircraft flying in close trail formation just below and aft of the aircraft carrying the RASCAL pod.

For the first ground test both the test aircraft (F-16D 90-00797) and the target aircraft (F-16D 87-00391) were parked on the TPS ramp separated by a distance of 100 meters. The RASCAL pod with the LADAR installed was mounted on the left wing on munition station 3 of the test aircraft. For the first ground test, an adjustable mirror assembly was positioned below the RASCAL pod viewing window such that laser pulses were projected parallel to the ground to view targets in front of the test aircraft. Figure 22 shows a schematic of the RASCAL pod, LADAR system, and adjustable mirror setup used to complete the tests, while Figure 23 shows a photograph of the entire system in operation.

In an attempt to increase the maximum range the LADAR could image, some modifications to the test setup were made for the second ground test. The second ground test used the same test aircraft, but the target aircraft was replaced by an un-painted MiG-15. The LADAR was still mounted in the RASCAL pod, however the pod was not mounted on the test aircraft, but was mounted sideways in its maintenance and storage rack, projecting the LADAR FOV parallel to the ground

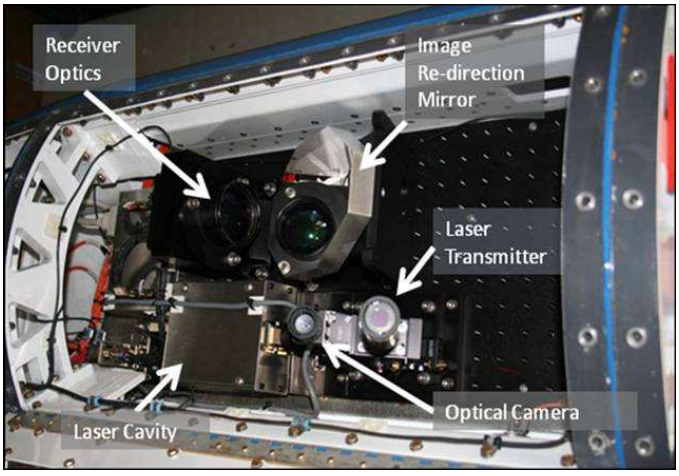

Figure 21: The LADAR configuration in the RASCAL pod [1]

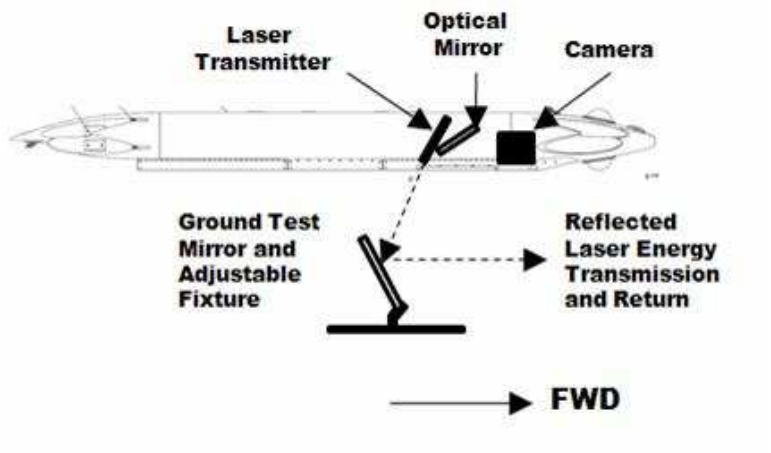

Figure 22: Schematic of the RASCAL pod, LADAR, and test mirror

Figure 23: The test aircraft loaded with the RASCAL pod

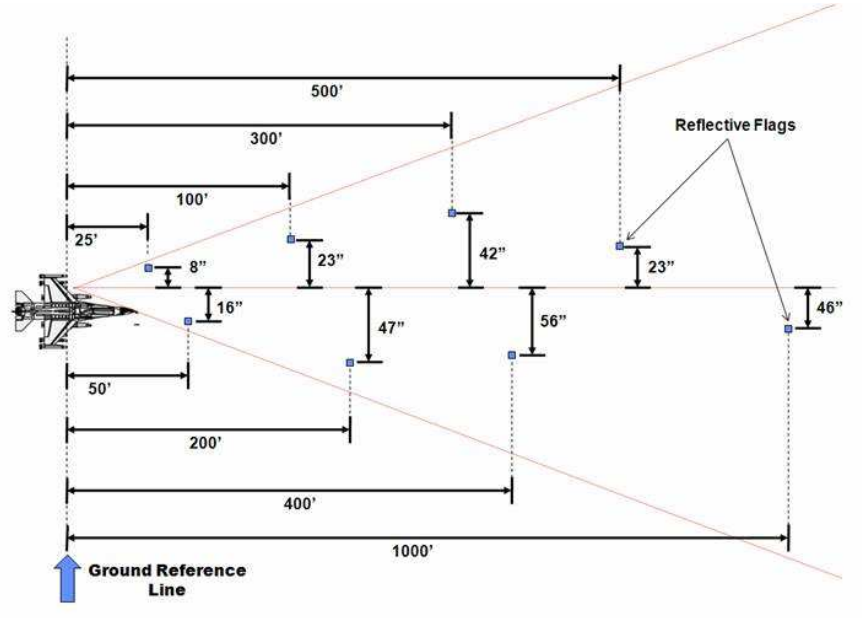

Figure 24: Schematic of the test layout

without use of the ground test mirror assembly. Additionally, the polycarbonate window was removed from the optical path of the laser transmitter and the LADAR receiver.

For both tests, an array of reflective targets were placed in front of the test aircraft at known positions and ranges along the ground. The reflective targets consisted of small wood blocks with dowel rods protruding from the blocks and a small reflective flag attached to the end of the dowel. The reflective flags were sized to minimize saturation of individual image pixels for later post-processing. Figure 24 shows a schematic of the reflective flag layout placed in front of the test aircraft for imaging. Figure 25 shows a photograph of the test layout including the reflective flag array and the target aircraft used for the first ground test. Figure 26 shows a typical LADAR intensity image of the target array captured during the first ground test, while Figure 27 shows a typical intensity image of the target array captured during the second ground test.

Figure 25: Photograph of the test layout

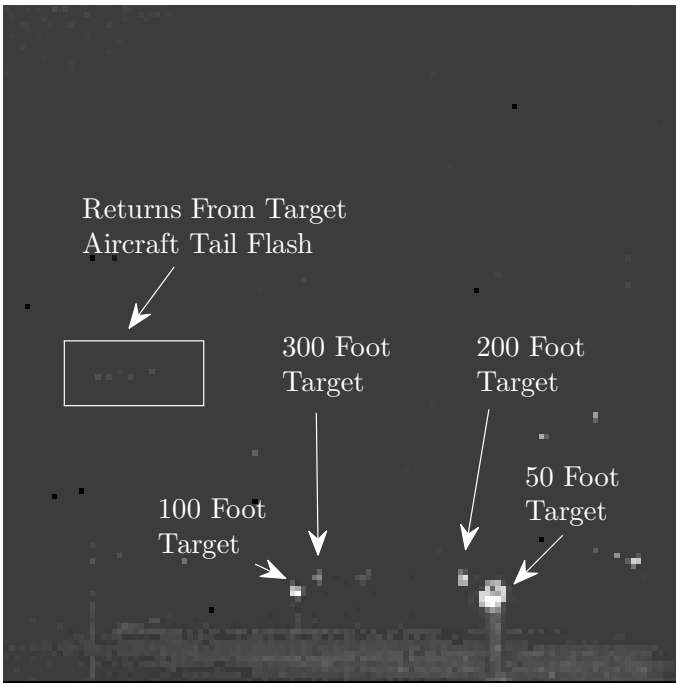

Figure 26: Typical LADAR intensity image from the first test

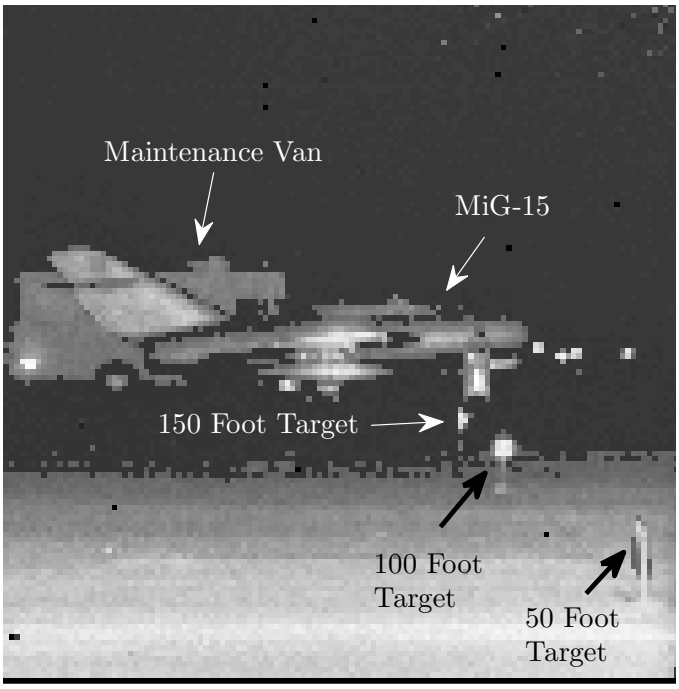

Figure 27: Typical LADAR intensity image from the second test

4.2 LADAR Data Collection

During both ground tests the LADAR was configured to record a 36 frame long sequence of three-dimensional images. During the first test 76 sequences were taken over the course of an hour with frame rates varying between 2, 5, 10, and 20 frames per second. The threshold was set at a constant value of 1.29 while the APD gain was set at 38.69. These settings are non-dimensional values specific to the ASC system. For the second ground test the frame rate was held constant at 10 frames per second during the collection of 21 more sequences. The threshold value was adjusted between 1.30 and 1.40 while the APD gain was adjusted between 35.0 and 42.0.

As can be seen from Figure 26, the first ground test images provided relatively few useable pixels per image. Only 19 pixels from the closer reflective targets and a reflective portion of the aircraft tail were consistently suitable for running any ranging algorithm. All other targets including the target aircraft had returns that were too weak to be detected by the LADAR perhaps due to lower than expected

laser output, low target reflectivity, or absorbtion, scattering, or reflection from the ground test mirror assembly or the RASCAL polycarbonate window. However, due to the number of files and frames taken, this still resulted in over 15,000 rangeable waveforms from those 19 selected pixels.

Much better images were captured during the second ground test possibly due to removal of the RASCAL window, removal of the ground test mirror assembly, or the highly reflective metal surface of the MiG-15.

4.3 LADAR System Effects

Post test review of the data collected during the two ground tests indicated that the ASC LADAR possessed several characteristics requiring adjustments to be made in the data reduction and analysis. The characteristics below were considered to be system effects of the ASC LADAR that affected the ability to estimate range. While some could be easily accounted for and removed prior to data analysis, others could not and injected a considerable amount of uncertainty into the range estimates.

4.3.1 Timing Offset and Dropped Slices. An unknown timing offset was applied to all recorded LADAR waveforms due to a delay between the generation of the outgoing pulse and the beginning of time measurement. Additionally, single pixels would occasionally miscount or *drop* a slice from the number of slices stored for the marker slice. An example of the latter is shown in Figures 28 through 30 which show three normalized LADAR returns from the same pixel across three subsequent frames. As time progressed from the first frame to the second, the intensity of the recorded laser pulse decreased enough to trigger the detector an entire slice later than the first frame. This is indicated by a one-slice-left shift of the recorded waveform and should have resulted in an increase in marker slice range by one slice. NOVAS estimated the peak to be at 37.4 slices in the first frame and 36.4 slices in the second frame because the marker slice position still reported 22 slices in the second frame when it should have increased to 23 slices. This equated to nearly 35 cm of error if operating at the

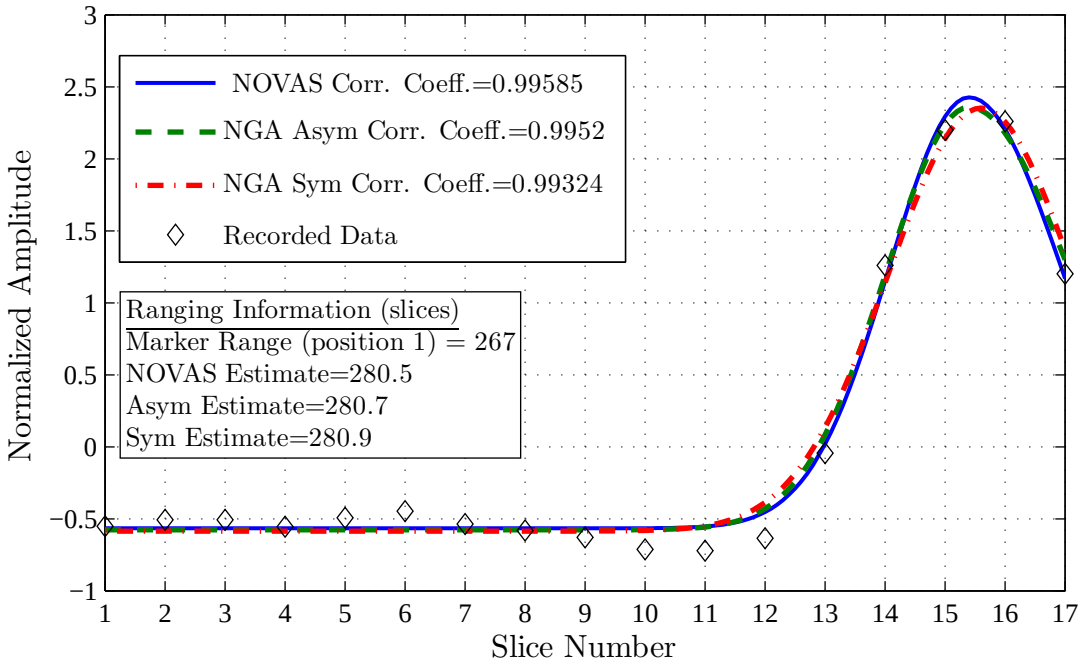

Figure 28: LADAR waveform one frame before a slice dropout event

nominal 400 MHz. In the third frame, the dropped slice was correctly added back to the marker slice range.

Dropped slices often accompanied a change in return pulse amplitude that moved the trigger point, but occasionally occurred with otherwise constant pulse characteristics. The dropped slices were entirely unpredictable, therefore the only way to detect them was to search for a series of samples that had characteristics such as those shown in Figures 28 through 30.

4.3.2 Clock Oscillator Error. Another source of uncertainty in range estimation came from the variation in the frequency of the system oscillator, often referred to as clock jitter. While the system was designed to record measurements at a nominal 400 MHz, this frequency was not consistent. This caused the length of time between recorded slices to vary. As time and distance are directly proportional via the speed of light, the physical range measured by a given number of slices varied also. The end result was that the range to a target at a fixed distance away from the LADAR

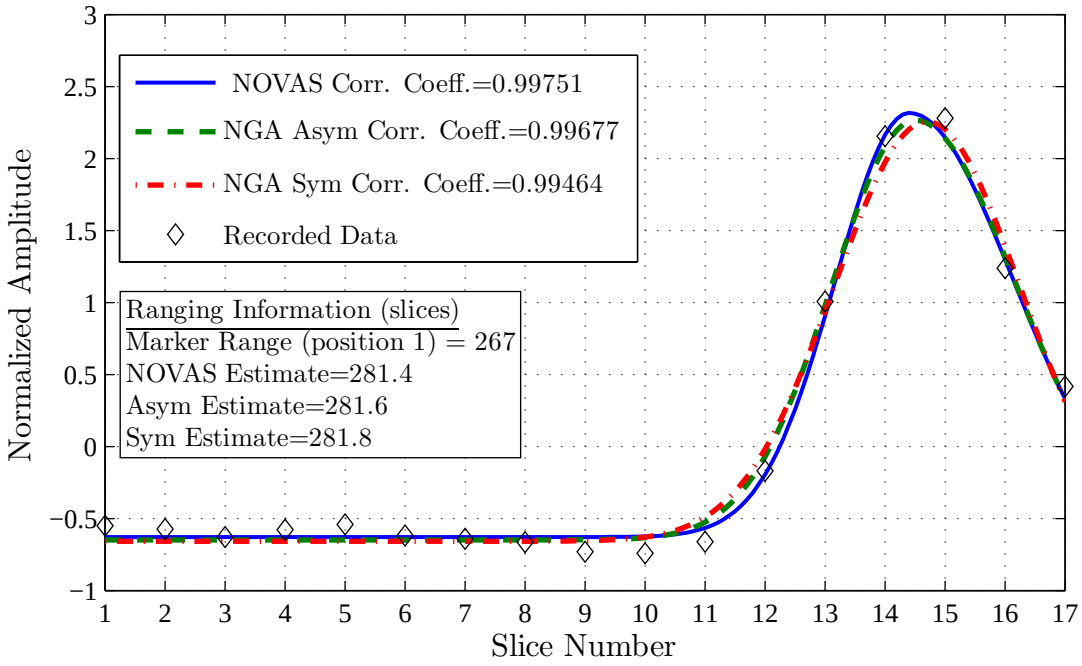

Figure 29: LADAR waveform during a slice dropout event

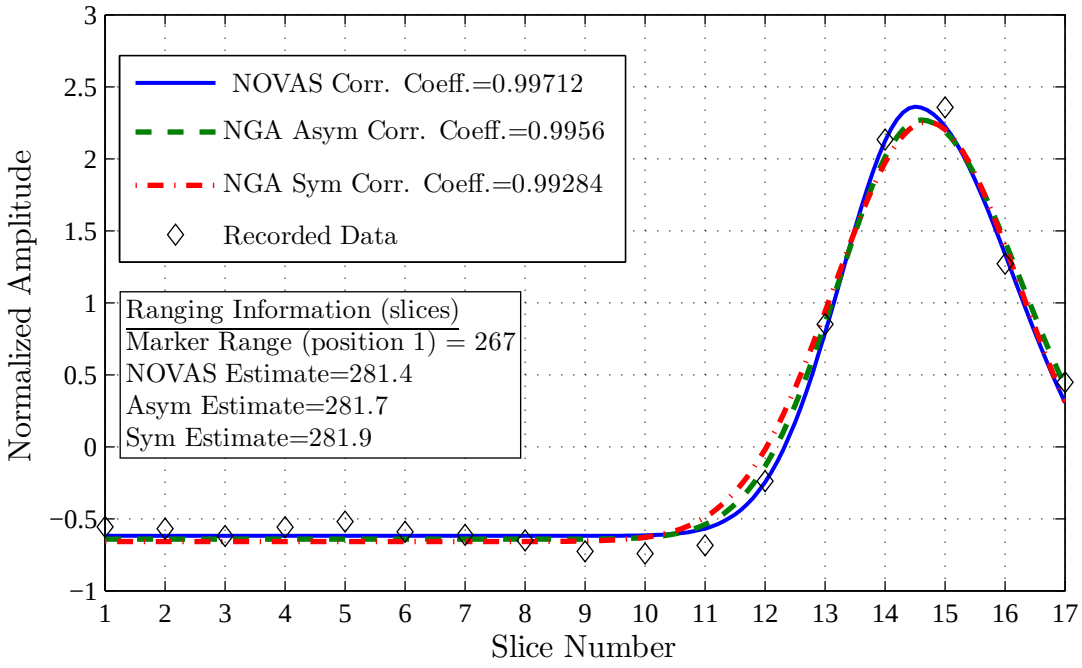

Figure 30: LADAR waveform one frame after a slice dropout event

was recorded as a varying number of slices depending on the clock frequency at the time. As there was no method in place to measure the clock frequency, the range represented by a given number of slices is also unknown. For this reason, the number of slices counted will be used to measure the recorded range for a large portion of this discussion.

All pixels of a particular image recorded at the same time were subject to the same shift in clock frequency. Thus on a frame by frame basis the range shift due to the clock frequency shift was applied equally to all pixels in that frame. Figure 31 illustrates this point using a single sequence of 36 frames taken during the second ground test. The target scene did not change through the course of the sequence, and an average range for each of 307 pixels used was found across all 36 frames. For each of the 36 frames, all the selected pixels had their corresponding average range subtracted from them, creating a zero-mean time history of range measurements for all 307 pixels through the 36 frames. These time histories were then plotted on the same axes. The bold black line depicts a zero mean representation of the average range to all pixels in a given frame. The horizontal axis depicts 3.6 seconds of time in the form of 36 frames taken at 10 frames per second. The vertical axis depicts the deviation from the average range of each pixel in terms of number of slices. Note that all the pixels follow the same range shift trend as time progresses due to the time dependant nature. To isolate clock error from slice dropout, slices with slice dropout were not used in calculating any of the averages discussed here.

4.3.3 Quality of LADAR Output. Several factors affected the ability of the ASC LADAR to record a waveform that was a true representation of the incident return pulse. Within a given frame of recorded slices, the values recorded immediately before and after the marker slice in the circular sequence were corrupted with an unknown negative bias pulling the values for that slice toward zero. Removing the first and last slices from the recorded waveform improved the results of all correlation techniques.

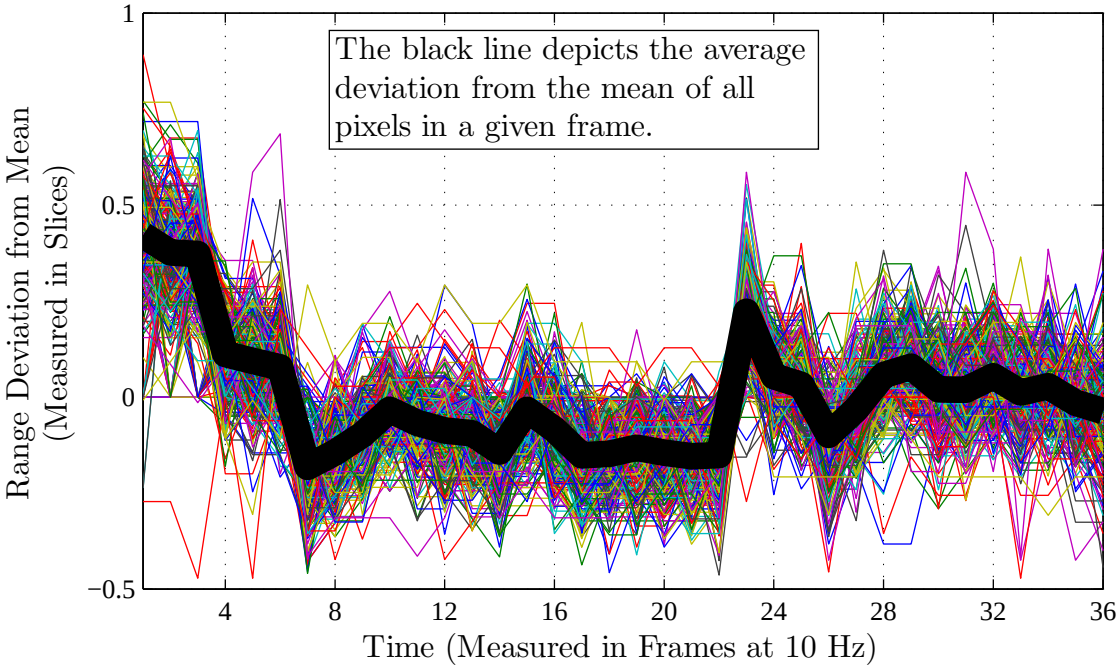

Figure 31: A collection of 307 zero mean pixels across 36 frames showing the error trends due to clock jitter

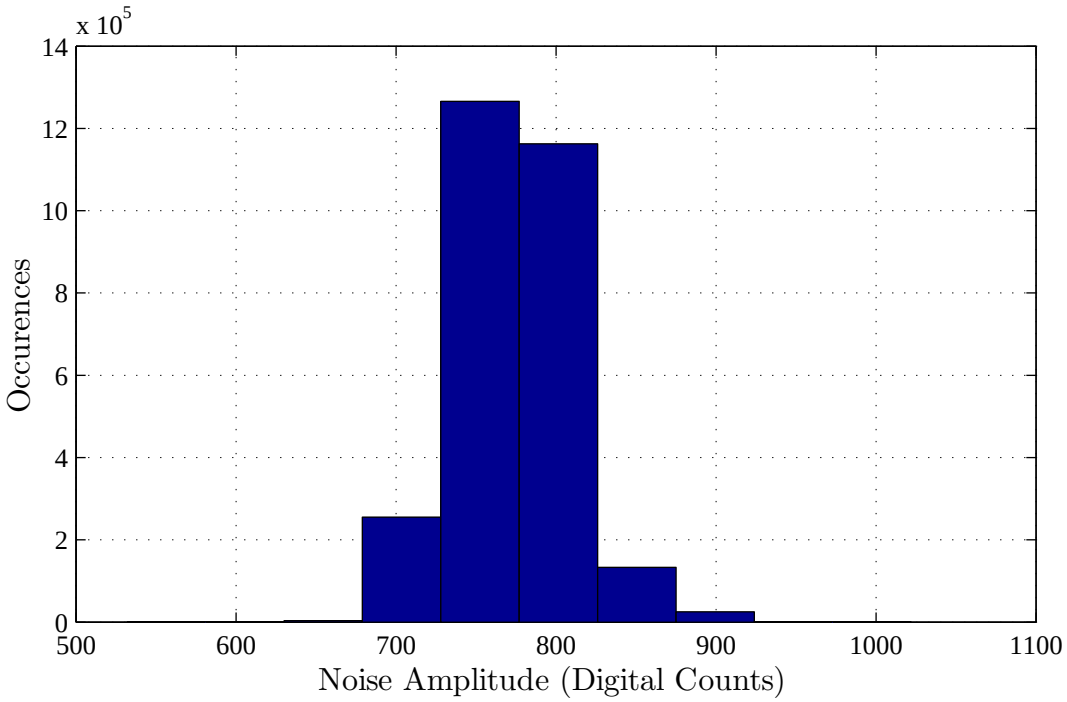

Figure 32: A histogram of recorded intensity of background radiation and noise

Additionally, the amplitude of the recorded LADAR return played a role in the quality of the recorded waveform. This amplitude was recorded in terms of a digital count representation of the electrical current output by the detector element for a given time slice. Possible values for the digital count ranged from 0 to approximately 3,500. The average value for a pixel that did not record a laser pulse return but only captured background radiation and noise was 771 digital counts with a standard deviation of 33 digital counts. Figure 32 shows a histogram of the background noise. Waveforms with an amplitude less than 1,000 digital counts were sufficiently deformed by the presence of noise that no models for the return pulse shape applied.

Furthermore, as the amplitude increased above 3,000 digital counts the detector element appeared to saturate and was no longer able to generate enough current to adequately respond to the incoming laser pulse. In these saturated cases, the stored waveform had a jagged plateau-shaped top and was not an accurate representation of the incoming laser pulse. In these saturated cases the estimated target range

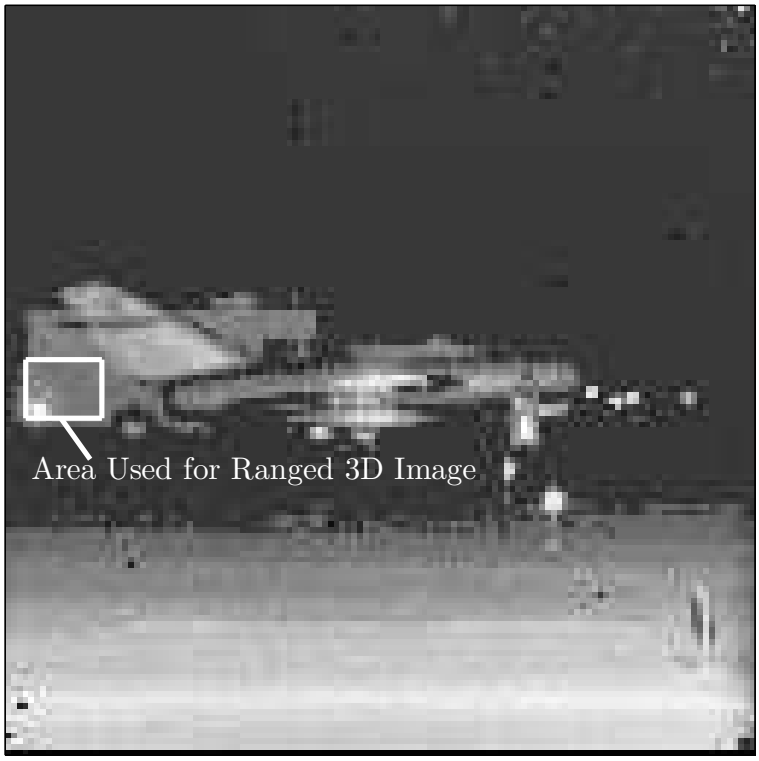

Figure 33: An intensity LADAR image showing the portion of the van used to discuss saturation effects on range estimation

was found to be shifted one to two slices closer than the actual target range. An example of this is shown in Figure 33 where the retro-reflective tail light of an aircraft maintenance van saturated the detector pixel. The rectangle in the image shows the section of the image used for illustration.

Figure 34 shows a three dimensional representation generated with the NOVAS algorithm in which the tail light appeared to protrude 2.3 slices from the rest of the flat surface of the vehicle. Assuming the oscillator was operating at its nominal frequency of 400 MHz this showed the tail light protruding over 80 cm when in fact it was flush with the rest of the vehicle.

Figure 35 shows two recorded (non-normalized) returns from the van section. One is from the saturated tail light the other is from a non-saturated portion of the van body. Also on the plot are the NOVAS estimations accompanying those recorded waveforms, re-scaled to match the non-normalized data. The third data set is a

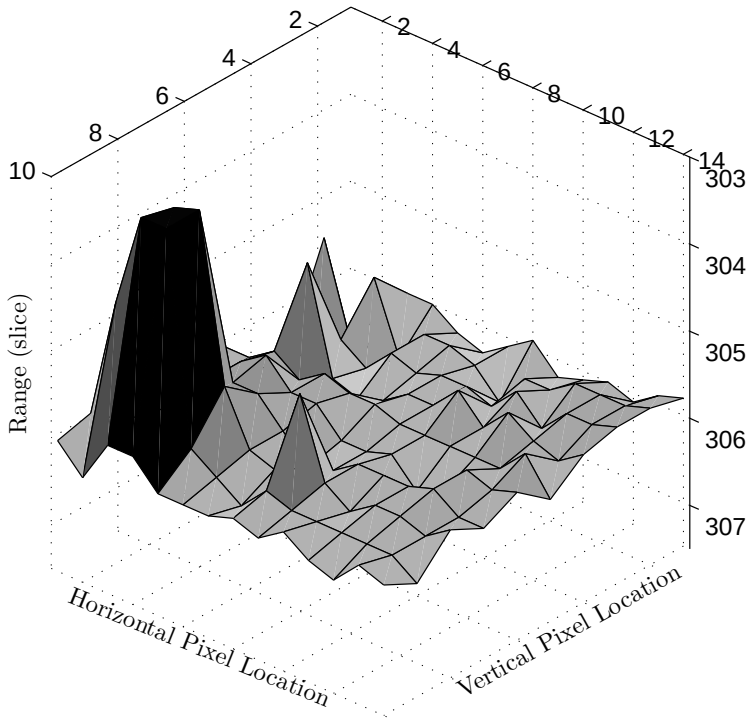

Figure 34: A three dimensional representation of the van section showing the erroneous range measurement due to pixel saturation

prediction of the true shape of the return pulse from the tail light . The prediction is based on the initial rise of the saturated pulse and the average range from the other 100 pixels from the van surface. This illustrates that the saturation cutoff amplitude has the effect of distorting the pulse shape and shifting the peak of the recorded waveform significantly further forward than the actual return pulse peak.

Since the purpose of this thesis was to determine the effects of laser pulse shape on range estimation algorithms, it was decided that only returns with undistorted pulse shapes would be used in the analysis. Returns with amplitudes between 1,000 and 3,000 were retained, which captured 50 percent of the pixels to imaged targets and provided over 250,000 waveforms for analysis from both ground tests.

4.4 Variation in Pulse Shapes

All 250,000 useable waveforms were run through the NOVAS estimator and found to have a significant variability in asymmetry and pulse widths. The reference

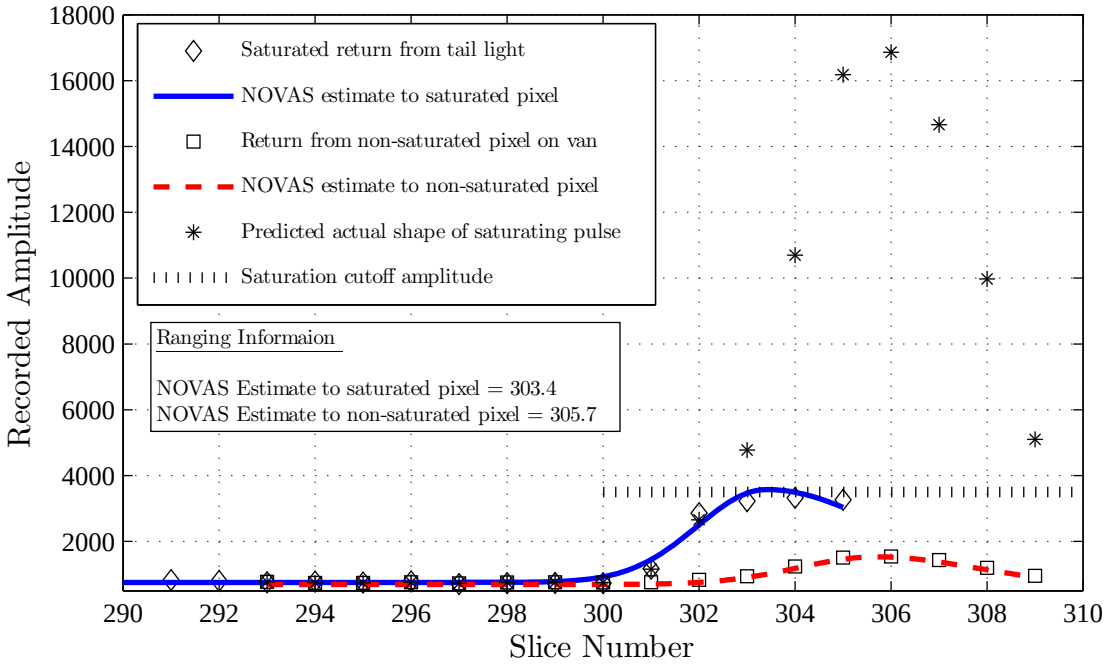

Figure 35: A comparison of saturated and non-saturated pixels from the van

matrix used contained 16,400 reference waveforms with range increments of 0.1 slices. Left half pulse widths available were from the set 0.61, 0.71, 0.81, 0.91, 1.02, 1.12, 1.22, 1.32, 1.43, 1.53, 1.63, 1.73, 1.83, 1.94, 2.04, 2.14 slices and right half pulse width available were from the set 0.91, 1.02, 1.12, 1.22, 1.32, 1.43, 1.53, 1.63, 1.73, 1.83, 1.94, 2.04, 2.14, 2.24, 2.34, 2.45 slices.

Table 1 shows the arithmetic mean, standard deviation, and mode of the half pulse width values for the left and right side of the waveforms. It can be seen that the pulses were clearly asymmetric with the right half of the pulse 25-30 percent larger than the left side. What was more significant was the difference between the shape of the laser pulses generated during the first ground test and those generated during the second test. The pulses were narrower during the first ground test than during the second. Additionally, the right half pulse widths showed more variance during the first ground test than during the second. Potential causes for this will be discussed in Chapter V.

Table 1: Pulse width statistics in terms of number of slices

Data Set	Mean (slices)	Std Dev (slices)	Median (slices)	Mode (slices)
Left Side, First Test	1.40	0.25	1.32	1.33
Left Side, Second Test	1.62	0.21	1.53	1.43
Right Side, First Test	1.69	0.34	1.63	1.53
Right Side, Second Test	2.04	0.17	2.04	2.04

Figures 36 through 39 depict histograms of the left and right half pulse widths. The bins across the horizontal axis represent each of the available pulse widths in the reference matrix, while the number of occurrences are shown on the vertical axis. It can be seen from these figures that in each case the half pulse widths are governed by similarly shaped distributions. The distributions themselves are asymmetric with the tendency for the pulses to more often vary wider than narrower.

The difference between the two ground tests is also visible from the figures. It is clear that the pulse width is dependent upon specific factors which cause the governing distribution to shift between the two ground tests.

However, the difference in variation of the right half pulse width is more an artifact of the NOVAS algorithm and the nature of the waveforms captured during the two tests. As mentioned earlier, the threshold was set at a constant, relatively low value during the first ground test. The result of this was that all the detectors triggered early in the initial rise of the pulse amplitude. Since the method of recording the data waveforms was fixed, an early trigger means less of the subsequent pulse will be recorded. Combined with the corruption of the last slice in the waveform, this caused a moderate truncation of the right side of the recorded waveforms and a predominate number of them had only two useable data points past the peak of the pulse. As discussed in Chapter III and as can be seen in Figure 20, NOVAS suffers from increased error in pulse width estimation under these conditions. This caused the stretching of the right hand tail of the distribution in Figure 38. Due to the increased threshold settings used during the second ground test, the waveforms

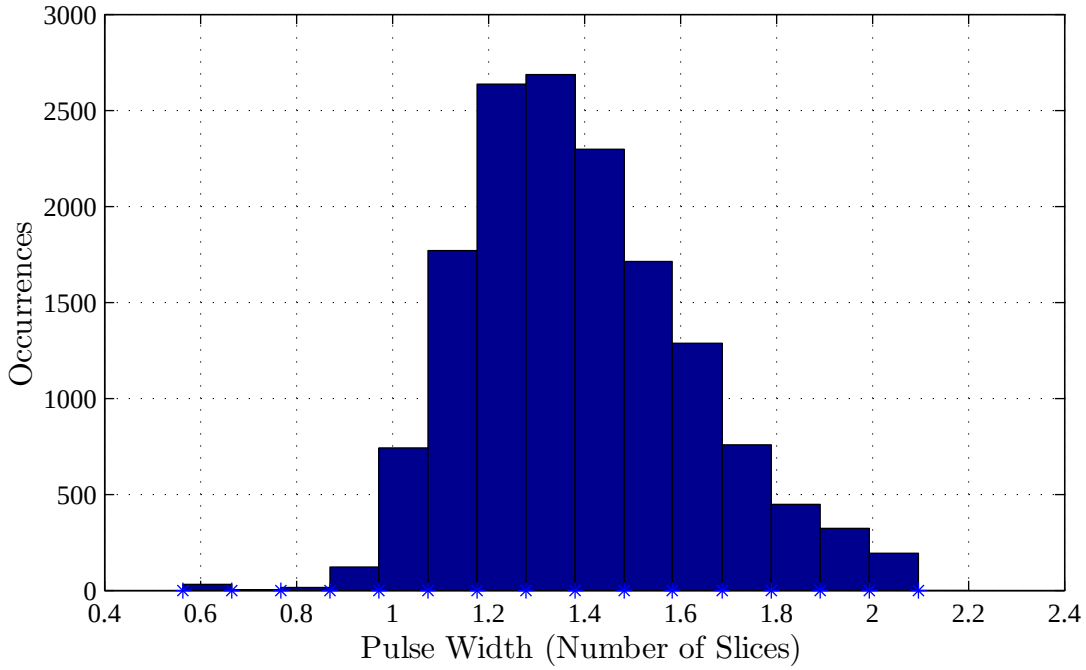

Figure 36: Histogram of left half pulse widths from the first ground test, estimated using NOVAS

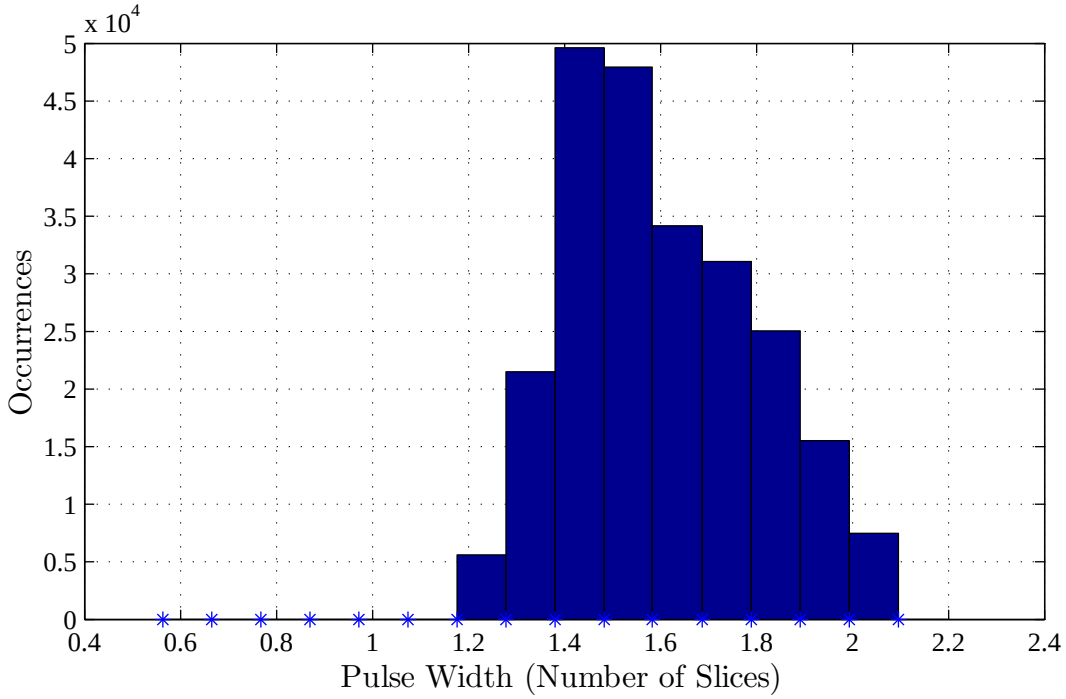

Figure 37: Histogram of left half pulse widths from the second ground test, estimated using NOVAS

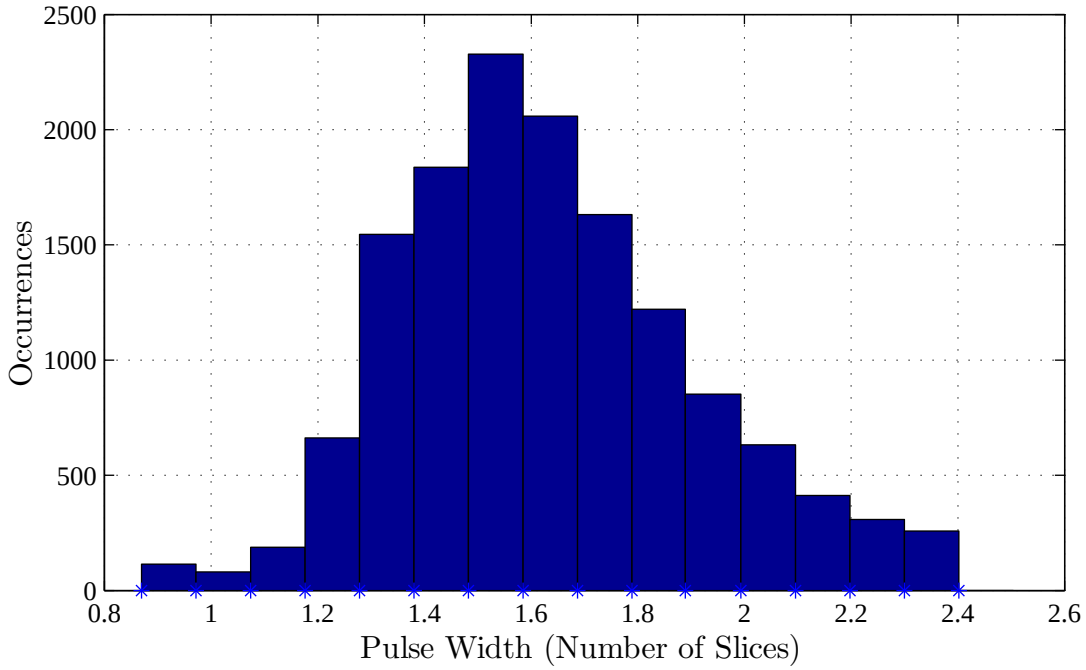

Figure 38: Histogram of right half pulse widths from the first ground test, estimated using NOVAS

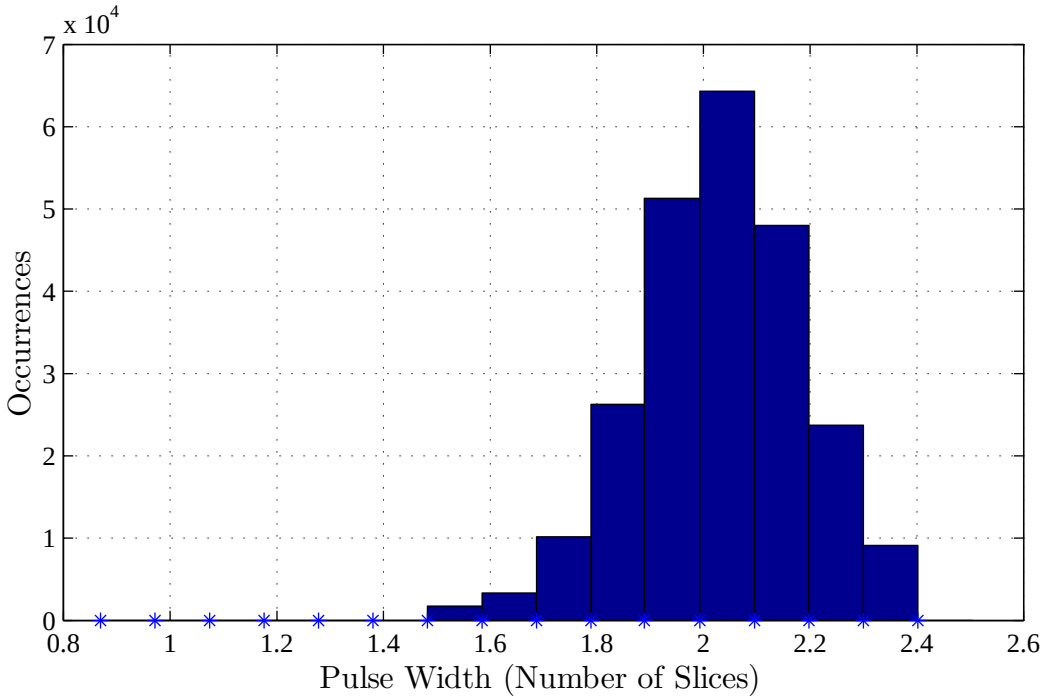

Figure 39: Histogram of right half pulse widths from the second ground test, estimated using NOVAS

collected then did not suffer as much truncation and the distribution in Figure 39 did not have as much stretching in the right tail.

4.5 Evaluation of NOVAS in Adaptation to Waveforms

The NOVAS algorithm performed similarly during the two ground tests as it did in simulation. All 250,000 waveforms meeting the criteria set forth in section 4.3.3 above were run through NOVAS and several other correlation algorithms for comparison. The reference matrix used by NOVAS was the same used to generate the pulse width analysis in section 4.4. The other algorithms used normalized data and reference waveforms in the same manner as NOVAS, but the reference waveforms were limited to a single pulse shape based on the averages shown in Table 1. These fixed shape waveforms are referred to as NOVAS generated averages (NGA) since the values that define their shape were determined by averaging the results of pulse shape correlation with NOVAS.

The purpose of this comparison was to first look at how NOVAS, which was allowed to change the shape of the reference waveform to match each individual data waveform, compared to a correlation that used the same reference waveform shape for all data waveforms. For this comparison, NGA waveforms were used as the fixed shape references such that NOVAS could be compared to well-shaped references. The goal was a direct comparison with emphasis on the difference between the range estimates rather than comparison to real-world range. This allowed a comparison free of the effects of clock error, delay bias, dropped slices, and individual pixel error. Comparison with an arbitrarily chosen waveform is contained in Chapter V.

For the first ground test, an NGA symmetric reference waveform was used with both left and right half pulse widths set to 1.55 slices. This was the average pulse width of all left and right half pulse widths combined from the first test. The NGA asymmetric waveform used had a left half pulse width of 1.39 and a right half pulse

width of 1.69, which were the averages of the respective halves from the first ground test only.

Similar NGA reference waveforms were used in the comparison algorithms for the second ground test, with the left and right half pulse widths equal to the averages in the same manner as with the first ground test. In this case the NGA symmetric reference waveform had both left and right half pulse widths set to 1.82 slices, while the NGA asymmetric waveform with a left half pulse width of 1.61 slices and a right half pulse width of 2.04 slices.

Figure 40 shows an example from the second ground test with the data points shown as diamonds. Also shown are the reference waveforms with the highest correlation values from NOVAS, the NGA asymmetric, and the NGA symmetric fixed waveforms. It can be seen that NOVAS achieved the highest correlation value. It can also be seen that while the correlation values for the fixed waveforms are not far below that of NOVAS, the fit of the reference waveforms to the data points does not appear to be as good as that of the NOVAS reference waveform. With that in mind, consider that a higher correlation does not necessarily mean a more accurate range. Range estimation accuracy will be discussed in a subsequent section.

The results from this open air test were remarkably similar to those found in simulation shown in Figure 12 in Chapter III. While all correlation values were slightly lower due to random noise in the signal, the relative correlation values were similar as were the relative positions of the three types of reference waveforms to the data waveform.

Figure 41 shows an example where NOVAS achieved a greater correlation difference over the other waveforms than in Figure 40. This may have been a case however where truncation of the right side of the pulse was reducing the NOVAS algorithm's effectiveness. In this example there were only two data points on the right side of the pulse. As shown in Figure 20 in Chapter III, this was a region where NOVAS began to introduce error into the right half pulse width estimation. Due to the low number

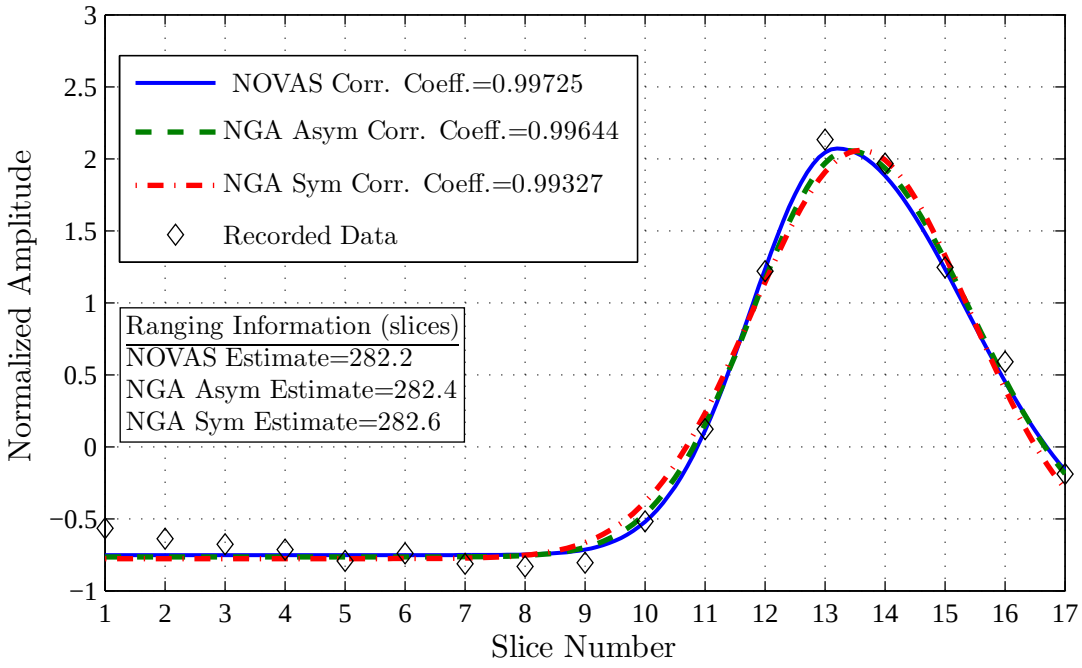

Figure 40: An example of the correlation achieved with the three different reference waveforms and a nearly complete data waveform

of points in the correlation of the right side, the selection of the right pulse width became increasingly susceptible to error due to variations from random noise. However the range estimation differences were consistent with those predicted by simulation, and the apparent asymmetry of the waveforms can easily be seen to be well matched by NOVAS.

Of the examined waveforms, 23 percent from the first test and 7 percent from the second ground test exhibited characteristics such as those shown in Figure 42. In this case the NOVAS chose a reference waveform with reversed asymmetry where the estimated left half pulse width is wider than the right. The result is a range estimation difference of 0.4 slices between NOVAS and the fixed asymmetric and 0.3 slices between NOVAS and the symmetric reference waveform. There are three possible conclusions to be drawn from this. First, NOVAS could be correctly adjusting for an anomalous waveform and be reporting an accurate estimation of the target range. Second, NOVAS could be incorrectly reacting to a waveform deformed by

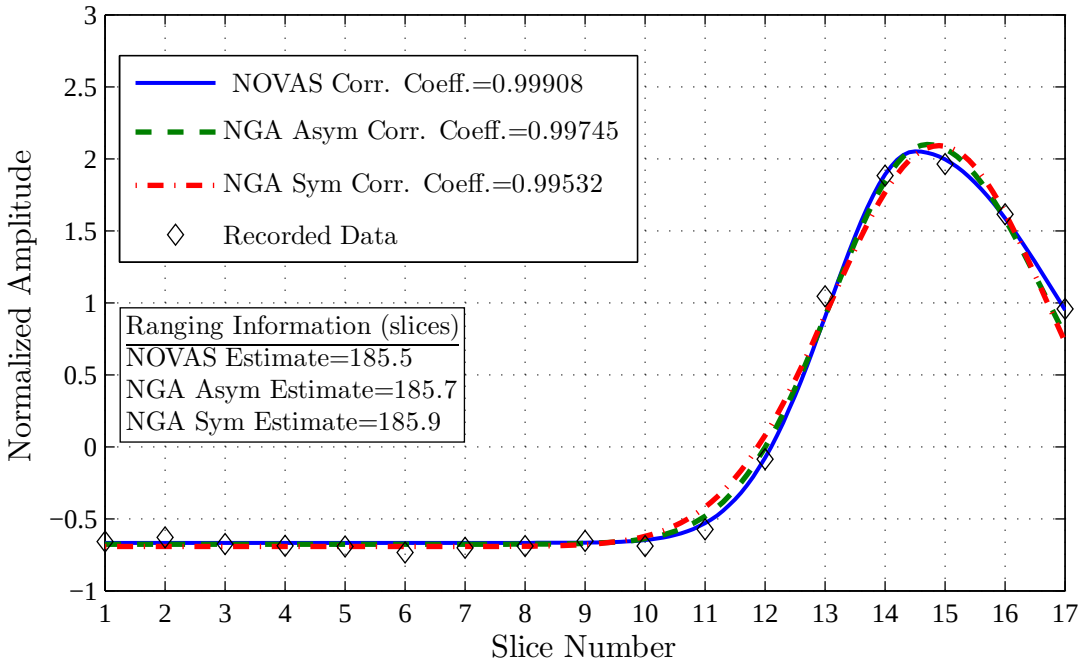

Figure 41: A comparison of the correlation achieved with the three different reference waveforms and a truncated data waveform

noise and reporting an incorrect range while the fixed reference waveforms are less affected by the noise and are providing a more accurate estimation of the target range. Thirdly, the data waveform may be so distorted that none of the references are able to report a correct range estimation. However the system errors of the LADAR system prevented any of the three conclusions to be definitively drawn.

Figure 43 shows a case where the fixed asymmetric reference waveform achieved a slightly higher correlation value than NOVAS. It just happened that this particular waveform was nearly the exact shape used for the fixed reference and fell between values available for NOVAS to choose for left and right pulse widths. The differences however were nearly imperceptible, and both references generated the same range estimation, at lease to within a tenth of a slice.

Figure 44 also shows a case where the fixed asymmetric reference waveform achieved a slightly higher correlation value than NOVAS. In this case the fixed asymmetric reference reported a range estimation two tenths of a slice less than NOVAS

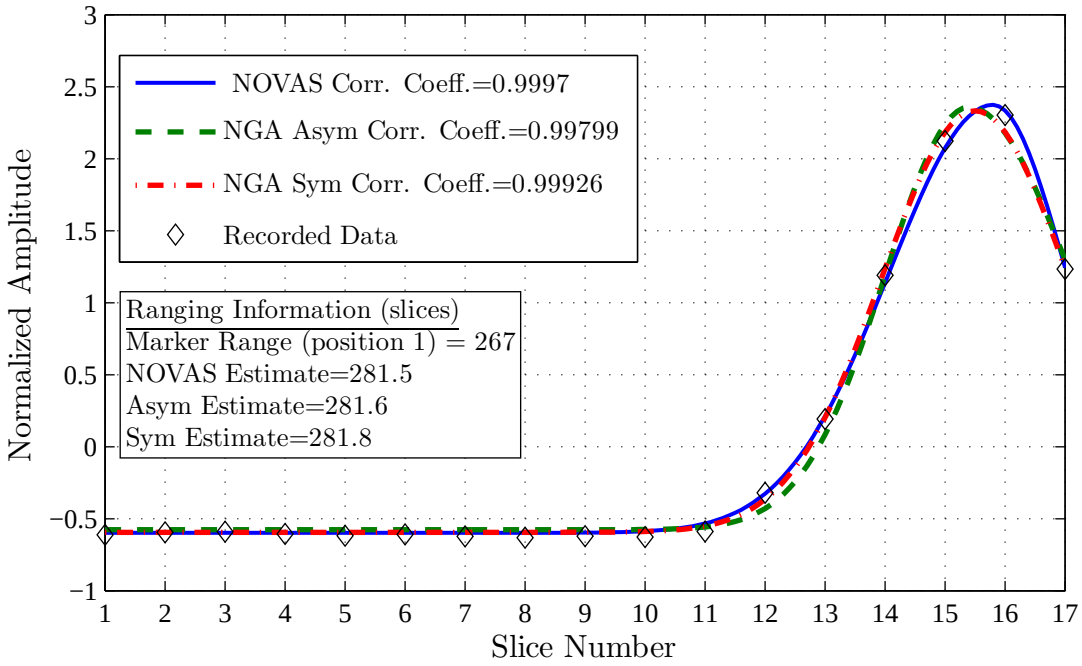

Figure 42: An example where NOVAS chose a wider left side than right in a rare occasion where the NOVAS range estimate was the largest of the three

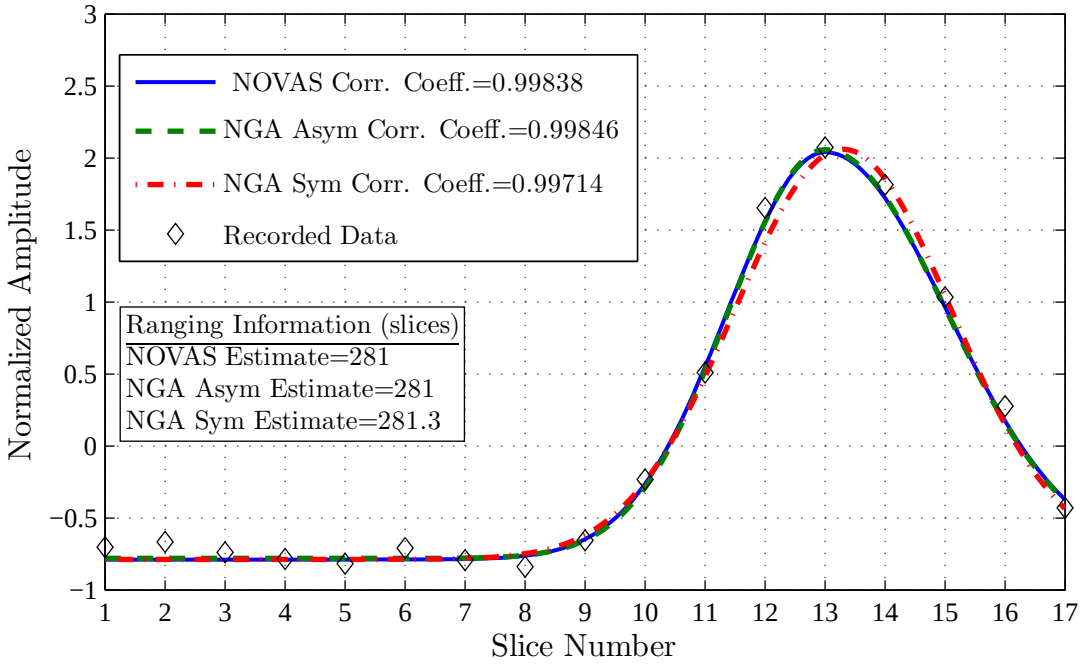

Figure 43: An example where the fixed asymmetric reference achieved a higher correlation but reported the same range estimation as NOVAS

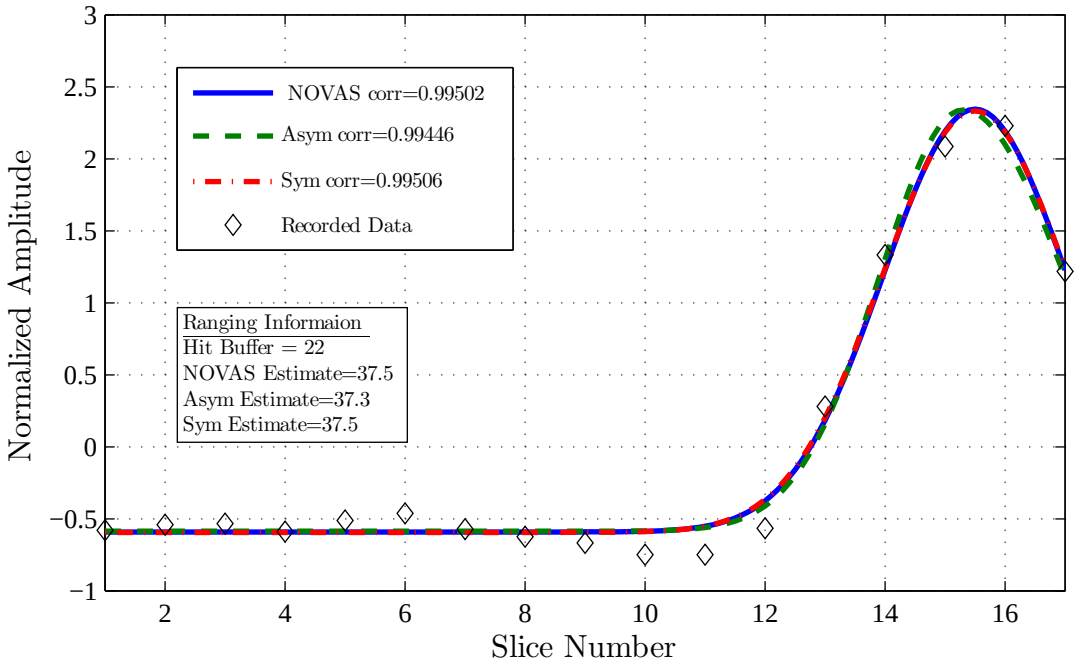

Figure 44: An example where the fixed symmetric achieved a higher correlation but reported the same range estimation as NOVAS

or the symmetric reference, which agreed on a range estimate to within a tenth of a slice.

The root mean squared difference between NOVAS estimates and the NGA fixed waveforms are shown in Table 2. The values are presented in number of slices as well as a range value assuming the system oscillator is operating at the nominal 400 MHz. This shows that NOVAS and the fixed asymmetric reference agreed on average of the total slice count to each pixel although a great variation existed on individual measurements. The symmetric reference consistently estimated a range slightly longer than NOVAS or the asymmetric reference. Variation between NOVAS estimates and the symmetric estimates were of similar magnitude as with the asymmetric reference. This is consistent with the simulation presented in Chapter III where the symmetric waveforms bias the range estimates long due to the wider back half of the asymmetric laser pulses.

Table 2: Range estimation difference between NOVAS and NGA fixed waveform algorithms in terms of number of slices and approximate centimeters

Algorithm	Mean Difference [slices (cm)]	RMS Difference [slices (cm)]	Maximum Difference [slices (cm)]
Asymmetric, First Test	-0.002 (-0.1 cm)	0.25 (8.8 cm)	0.60 (21 cm)
Symmetric, First Test	0.17 (6.0 cm)	0.27 (9.5 cm)	0.80 (28 cm)
Asymmetric, Second Test	0.00 (0.0 cm)	0.17 (6.0 cm)	0.50 (18 cm)
Symmetric, Second Test	0.21 (8.6 cm)	0.27 (9.5 cm)	0.80 (28 cm)

4.6 Comparison of Calibrated Range Estimates

A comparison of calibrated range estimations was performed accounting for all the factors discussed thus far. To make one set of comparisons three targets were used from the first ground test: the 50 foot target (15.24 m), the 100 foot target (30.48 m), and the 200 foot target (60.96 m) shown in Figure 26. The range to each of these targets was converted to meters for this analysis to be consistent with the simulation and was measured to within 0.005 m. From the entire set of 76 sequences of 36 frames, 144 individual frames were found in which the returns from each pixel met the desired criteria set forth in section 4.3.3. All dropped slices that could be manually detected were removed. For a second range comparison, a similar set of 117 frames were selected based on returns from a 201 foot target and a 202 foot target not shown in Figure 26. These targets were 0.34 m apart.

Range estimates were generated using the NOVAS algorithm with the same available pulse shapes used for the analysis in section 4.4 above. These estimates were compared to those generated using the fixed-shape NGA asymmetric waveform for the first ground test determined using the averages in Table 1 as well as the fixed-shape NGA symmetric waveform with an average overall pulse width also determined from the averages in Table 1. Several different approaches to system calibration were used to counter the effect of the systematic errors.

4.6.1 Calibration Procedures. The calibration method used for accounting for the systematic errors such as timing offset, dropped slices, or clock frequency required two targets of known range from the LADAR to be captured in the LADAR image. The clock frequency was estimated by comparing the number of recorded slices between the two known-range targets to the actual range between the targets. This determined the range value per slice, which through the relationship to the speed of light gave the time value per slice or clock frequency. This value was then used to compare the total number of slices recorded to each of the targets to the known range to each target to determine the timing offset and to detect dropped slices.

Unfortunately, these calibration values were only good for that particular frame as the clock frequency, timing offset, and dropped slices all varied with time. Furthermore, there was additional system error that occurred on a pixel-by-pixel basis that affected the calibration values. While the source of this error is not investigated in this thesis, its presence can be seen in following figures. Two approaches to calibration were used to counter these two error sources.

To counter the clock frequency variation, three individual calibrations were performed for each of the 144 frames. The first used the 15 m and 30 m targets to calibrate range estimates to the 60 m target. The second used the 15 m and 60 m targets to calibrate range estimates to the 30 m target. The third used the 30 m and 60 m targets to calibrate range estimates to the 15 m target. The pixels used will be referred to as the calibration pixels and the target pixel.

To counter the pixel-by-pixel error effect on calibration, average calibration values were determined using every combination of the three calibration pixels across all 144 frames gathered. The average clock frequency and timing offset were then used to recalculate estimates to the 15, 30, and 60 m targets. All three waveform types agreed that the average clock frequency was 405.8 MHz. The fixed references both produced a standard deviation of 3.26 MHz while NOVAS produced a standard deviation of 4.08 MHz.

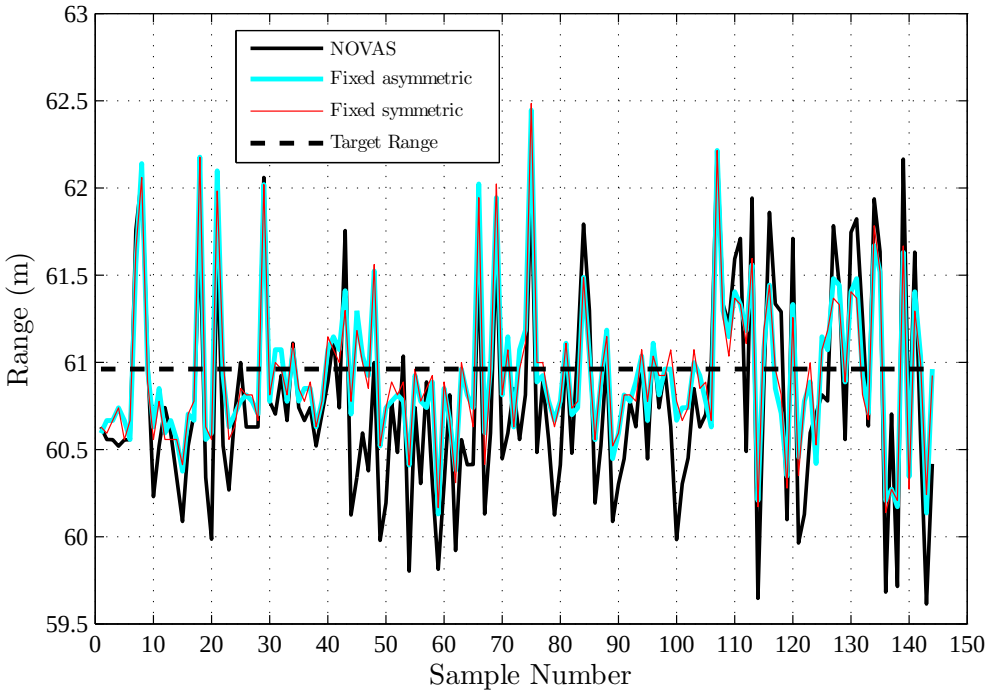

Figure 45: A comparison of range estimations to a target at 60.96 meters using three types of reference waveforms and a separate calibration for each sample

4.6.2 Range Estimations. The results across all 144 selected frames are shown in Figures 45 through 47. The 144 samples were not necessarily sequential in that some frames existed between those shown that had been thrown out for poor waveform quality. However, they are shown in chronological order. In these figures it can be seen that in each, all three types of reference waveforms follow the same general pattern of range estimations. The errors captured in these figures are manifestations of the individual pixel error to which each of the reference waveform types respond similarly. However the error shown is not only that of the target pixel. The errors in the calibration pixels are absorbed into the calibration values and projected upon the target pixel. Range estimates to both calibration pixels generated using the calibration values had zero error in every case regardless of the pixels used.

The NOVAS algorithm, however, appeared more affected by the pixel error. This is evidenced in the figures by the NOVAS results reaching higher extremes of positive and negative estimation error. In most cases the error for all three algorithms was in

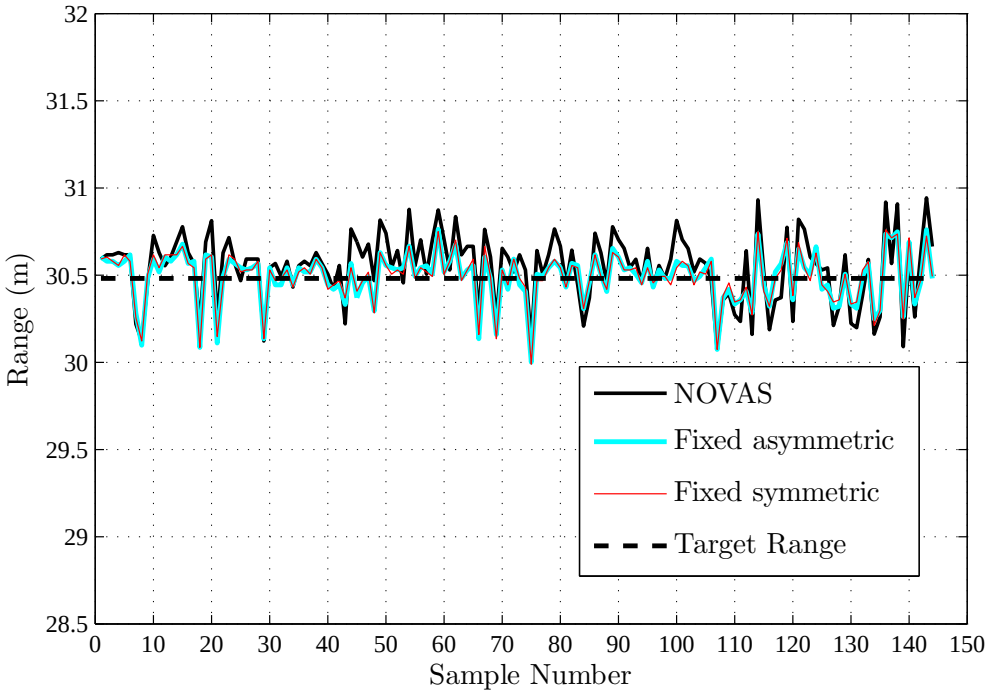

Figure 46: A comparison of range estimations to a target at 30.48 meters using three types of reference waveforms and a separate calibration for each sample

the same direction, however there were a few instances where NOVAS responded in an opposite manner.

The results of range estimates generated with the average calibration are shown in Figure 48 where the bottom panel is the estimate to the 15.24 m target, the middle panel is the estimate to the 30.48 m target, and the top panel is the estimate to the 60.96 m target. Here it can be seen that the magnitude of the step-by-step variation was less. All the pixel errors were now distributed to the individual pixels and not projected onto one pixel. The variation of the clock can also be seen by the increasing value of the range estimations as time progressed. Any error in the clock frequency calibration value manifested as a percentage error of the target range. This generated a larger numerical range error for the longer range targets.

The results of all range estimates are contained in Tables 3 through 5 including the mean estimation error from the known range, the standard deviation of the range estimates, and the root mean squared error (RMSE) of the range estimates. In all

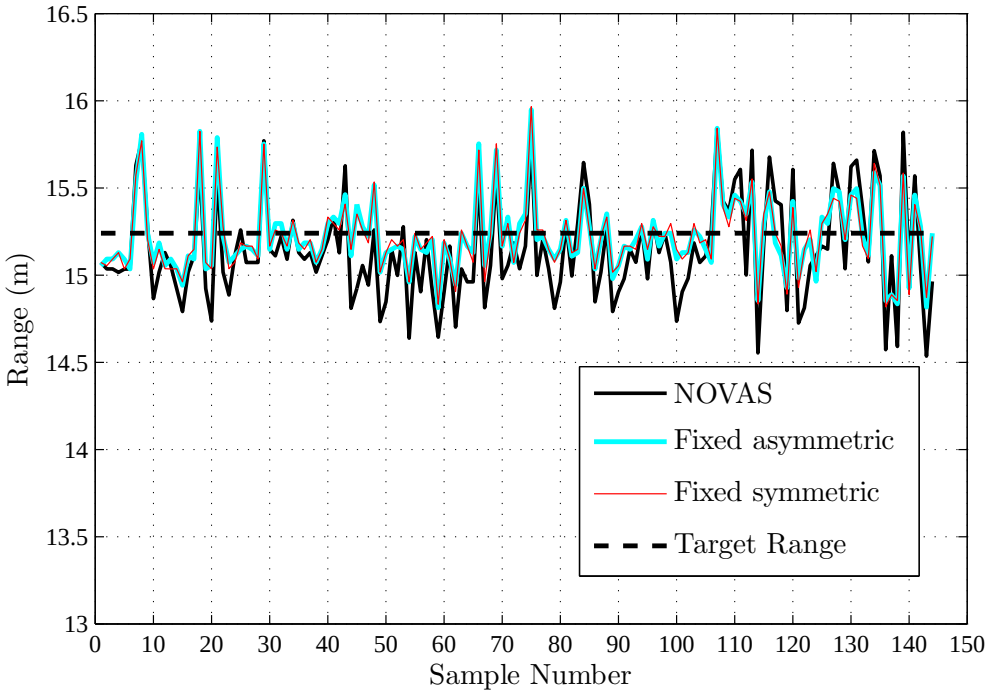

Figure 47: A comparison of range estimations to a target at 15.24 meters using three types of reference waveforms and a separate calibration for each sample

cases NOVAS generated five to ten times more error with nearly twice the variation of the fixed waveforms. On the other hand, the NGA fixed asymmetric reference generated the smallest error in five of the six cases, losing to the NGA symmetric reference in the average calibration estimate to the 60.96 m target. The fixed asymmetric reference even achieved zero error in one instance. The fixed asymmetric reference's standard deviation and RMSE were only slightly larger if not equal to that of the fixed symmetric.

Table 3: Results of range estimations to a 60.96 meter target

Method	Mean Error (m)	Std Dev (m)	RMSE (m)
NOVAS, single calibration	-0.180	0.579	0.605
NGA Fixed Asym, single calibration	-0.025	0.434	0.433
NGA Fixed Sym, single calibration	-0.029	0.430	0.430
NOVAS, average calibration	-0.122	0.250	0.277
NGA Fixed Asym, average calibration	-0.006	0.249	0.248
NGA Fixed Sym, average calibration	0.001	0.249	0.248

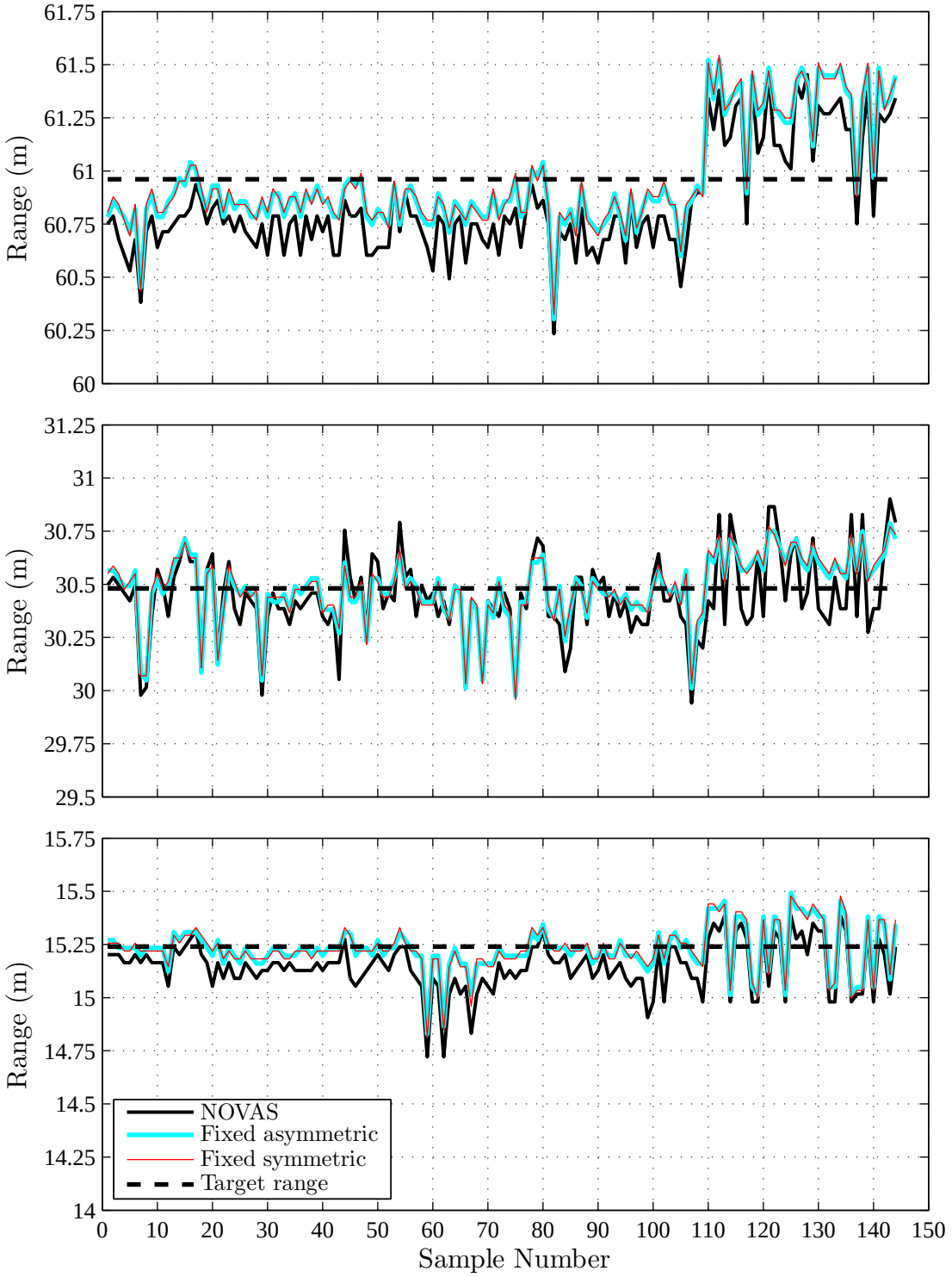

Figure 48: A comparison of range estimations to targets at 15.24, 30.48, and 60.96 meters using three types of reference waveforms and a single average calibration

Table 4: Results of range estimations to a 30.48 meter target

Method	Mean Error (m)	Std Dev (m)	RMSE (m)
NOVAS, single calibration	0.064	0.193	0.203
NGA Fixed Asym, single calibration	0.010	0.143	0.143
NGA Fixed Sym, single calibration	0.011	0.142	0.142
NGA NOVAS, average calibration	0.040	0.189	0.192
NGA Fixed Asym, average calibration	0.000	0.157	0.156
NGA Fixed Sym, average calibration	0.004	0.156	0.156

Table 5: Results of range estimations to a 15.24 meter target

Method	Mean Error (m)	Std Dev (m)	RMSE (m)
NOVAS, single calibration	-0.098	0.290	0.305
NGA Fixed Asym, single calibration	-0.016	0.214	0.214
NGA Fixed Sym, single calibration	-0.018	0.212	0.212
NOVAS, average calibration	-0.095	0.117	0.150
NGA Fixed Asym, average calibration	-0.011	0.108	0.108
NGA Fixed Sym, average calibration	-0.012	0.110	0.111

Table 6: Results of range estimations to targets separated in range by 0.34 meters

Method	Mean Error (slices)	Std Dev (slices)	RMSE (slices)
NOVAS	0.057	0.690	0.615
NGA Fixed Asymmetric	-0.045	0.468	0.485
NGA Fixed Symmetric	-0.055	0.469	0.596

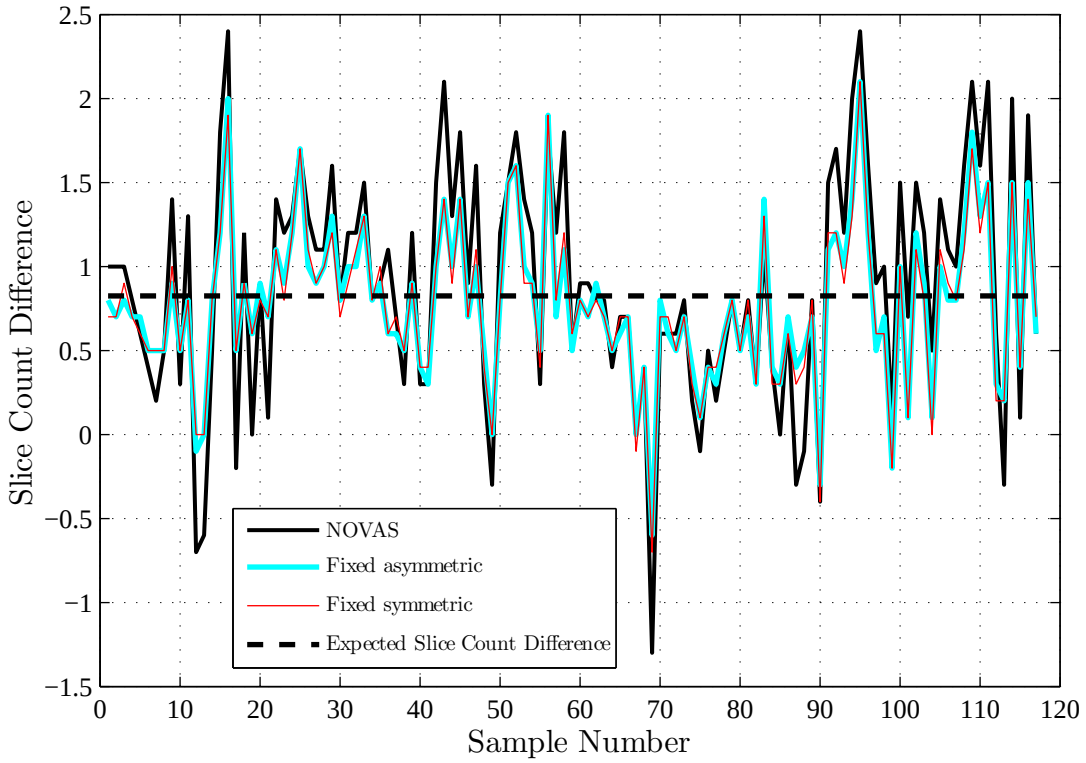

Figure 49: A comparison of slice counts generated for two targets separated by 0.304 meters using three types of reference waveforms

Furthermore, the NGA fixed asymmetric reference demonstrated a larger advantage when the distance between targets was shortened. The targets at 201 and 202 feet were used to accomplish this. The total number of slices estimated by each of the reference types were gathered from 117 frames containing well shaped waveforms to both pixels. Based on the average clock frequency of 405.8 MHz, the expected difference between the two targets was 0.825 slices. The results are shown in Figure 49. Table 6 shows that the magnitude of the range error difference between NOVAS and the fixed wavforms decreased significantly. Additionally, the margin between the fixed asymmetric and the fixed symmetric waveforms widened to the advantage of the fixed asymmetric reference. While the advantage is slight, the fixed reference waveform with an optimized asymmetric shape more accurately estimated the target range 6 of 7 times in all.

V. Conclusions

5.1 System Errors

System characteristics intrinsic to the ASC LADAR were found to be the largest source of error in range estimations. Time dependent variation of the system's clock frequency produced variations in range measurements on the order of 35 cm. When sampled at 10 Hz, the pattern of the clock variation as shown in Figure 31 appeared to be governed by a zero mean random process making it impossible to predict. Averaging a collection of pixels across time can generate a more consistent result but may not necessarily be correct and would not apply to a moving target. The system clock is the basis for generating all range estimates. Without any method of monitoring the clock frequency there was no way to remove the uncertainty in range measurements generated by the ASC LADAR unless two targets of known range are within the LADAR FOV.

The tendency to drop slices also affected range estimation accuracy by subtracting on the order of 35 cm without indication or warning. This occurred on nearly five percent of recorded waveforms and was also impossible to predict or detect in real time. Comparing the estimated ranges to those before and after for the same pixel could help recognize these errors which could then be corrected by reinserting the dropped slice. Dropped slices were correctable after the collection of several range estimates for the same pixel, assuming the target was stationary in the LADAR image.

High amplitude returns from either close-range or reflective targets that saturated the receiving pixel generated range errors comparable to clock error and dropped slices combined. While such returns were easily identifiable, the range errors were not recoverable as shape of the original return pulse was lost. With saturated pulses, the best case was to simply throw the range estimate out.

5.2 Source of Pulse Shape Variance

One discovery made with NOVAS was that the shape of the laser pulses used by the ASC LADAR varied noticeably. The variation causing the shape of the distri-

butions seen in Figures 36 through 39 could have been due in part to random noise added to the signal or actual variations to the shape of the signal pulse itself in accordance with the discussion in Chapter 2. Most surprising was the difference between the distributions of the pulse shapes from one test to the next. Both distributions had similar shapes and standard deviations, but the pulses from the second test were somewhat wider than those from the first test. The means from both the left and right sides were shifted to the right. The cause for this was most likely not random noise, but a change in the signal shape generated by the LADAR. Factors of the test setup could have played a role in this difference.

The most likely factor affecting the shape of the pulses was the operating temperature of the system. As described in section 2.1, the conditions of a Q-switched laser cavity will affected the shape of the generated pulse. It is plausible that the operating temperature of the LADAR could have affected the laser cavity in many ways including power available to the laser pump diodes, the pumping efficiency of the diodes, the maximum achievable population inversion level, cavity efficiency, and mechanization of the Q-switch itself. Several factors could have led to different operating temperatures.

The biggest contributor to operating temperature was likely the environmental conditions at the time of the test. The first test took place during the day when the ambient air temperature was nearly 100 degrees Fahrenheit and the surface of the RASCAL pod was subject to heating by solar radiation. To counter this, the surface of the the RASCAL pod was cooled convectively with an external air conditioner providing a large volume of air at approximately 45 degrees Fahrenheit. The second test was performed at night when outside air temperatures were approximately 70 degrees Fahrenheit and the air conditioner was not required. For both tests the RASCAL pod was subject to internal heating from the electric and electronic components contained within it. While temperature probes existed to monitor the internal air temperature of the RASCAL pod, the mechanism to record the temperature data malfunctioned during both tests providing no information on the actual environment surrounding the

LADAR system. As such it was not possible to determine whether the environment of the first test was hotter or colder than that of the second test, given the unknown variables of the air temperature, heating sources, and cooling sources. However, external environment was not the only factor affecting the operating temperature of the LADAR.

Different modes, methods, and time of LADAR operation could affect the amount of power used and the time available for cooling. During the first and second ground tests, systematically monitoring time intervals was not determined in the test plan to be an objective. Thus these parameters were subject to the pacing required to accomplish the planned objectives and were not controlled. During the first test, 76 sequences of 36 frames were collected over the course of 70 minutes, however the data collection occurred after power had been applied to the RASCAL pod for two hours during system troubleshooting. During the second test only 20 sequences of 36 frames and one sequence of 108 frames were taken over 75 minutes after power had been applied for only 45 minutes. During both tests, intervals between sequences ranged from 20 seconds to 15 minutes. Frame rate was held constant during the second test at 10 Hz while it varied from 2 Hz to 20 Hz at various times during the first test. Frame rate had a direct impact on the temperature of the laser pulse generating hardware as a higher frame rate drew more power with a shorter interval for conductive cooling. All of these factors had compounding, canceling, or interrelated effects on the LADAR environment.

The contribution of each factor to the LADAR operating temperature could not be adequately characterized from the results of the tests performed in support of this thesis. The only conclusion that can be drawn is that one or more of these factors contributed to a change in the shape of the laser pulse generated by the ASC LADAR. The extent of the variation possible has yet to be explored.

5.3 Effect of Waveform Capture

Certain features of LADAR waveform recording and storage could enhance the capability of NOVAS. As can be seen from examples such as Figure 28, the ASC LADAR system often unnecessarily truncated the right half of the waveform. Capturing the 12 slices prior to the trigger slice resulted in the storage of several slices of background radiation values that changed the value of the Pearson's correlation coefficient without contributing to the accuracy of the range estimation. The cost of capturing these unnecessary slices was a decreased number of slices available for correlation on the right half of the pulse and an increased sensitivity to random noise during pulse shape estimation.

A technique used to counter this effect on the second ground test was to raise the threshold in an attempt to trigger the pulses later in their amplitude rise. This gained one or two more slices on the right half of the pulse but caused those images with a lower threshold to be of poorer quality. Much fewer image details were captured because lower intensity returns failed to break the higher threshold and trigger the respective pixels. While this method was suitable for pulse analysis and generated slightly better waveforms for NOVAS to process, it greatly reduced the effectiveness of the LADAR and was not suitable for operational use.

5.4 Evaluation of NOVAS

In light of the variation seen in the shape of laser pulse generation, it was clear that a ranging algorithm that can adjust to the shape of the pulse was beneficial. NOVAS achieved a higher correlation than the NGA fixed asymmetric waveform 97.2 percent of the time during the first test and 98.8 percent of the time during the second test. NOVAS achieved a higher correlation than the NGA fixed symmetric waveform 99.1 percent of the time during the first test and 99.6 percent of the time during the second test. Many times NOVAS achieved a Pearson's correlation coefficient

of 0.999—a feat never accomplished by the other reference waveforms. However, maximum correlation value did not necessarily translate to range accuracy.

Range estimation with NOVAS produced a consistently different result than the NGA waveforms. Figure 50 shows the range returns from a single pixel across 36 frames from a single sequence taken during the second ground test. The target was stationary, so any variation in the range measurement was due to any of the sources of range uncertainty mentioned above. It can be observed that the results from all three reference waveforms followed the same general trends indicating the overall source of error was independent of the reference waveform used. It can also be observed that the lines generated by all three references maintain roughly the same position relative to each other. This indicates that the range differences shown in figures such as Figure 40 remain fairly constant for a given pixel across a given sequence of frames.

Figure 48 also shows the range difference, indicated by the downward shift of the NOVAS range estimates. This time, the luxury of having three precisely ranged targets allowed an accurate calibration to be accomplished. This showed that the NOVAS estimate was actually consistently incorrect. The variability of the pulse shapes resulted in a bias of range estimates short of the target range. Interestingly, the difference between the range estimates using the average calibration was on the same order of the differences calculated in Table 2.

The true benefit of of NOVAS lies in the pulse shape estimation capabilities rather than the range estimation capabilities. The reference waveform that achieved the best overall performance was the fixed asymmetric pulse shape generated using the average of pulse shape estimations from NOVAS. The results in Chapter IV compared the NGA asymmetric waveform with an NGA symmetric waveform with the asymmetric waveform giving a more accurate estimate six of seven times. However, when compared to an arbitrarily chosen symmetric waveform the results were much more significant.

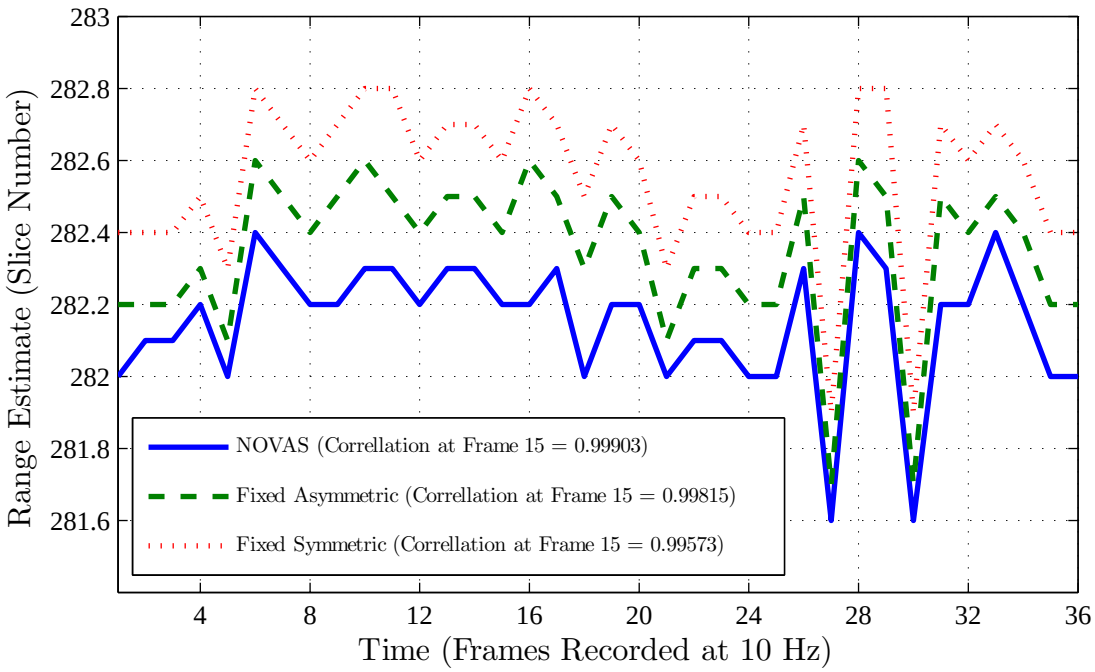

Figure 50: A comparison of range estimations from NOVAS, a fixed asymmetric reference, and a fixed symmetric reference (note the dropped slice events in frames 27 and 30)

Table 7: Comparison of results between the NGA asymmetric and an arbitrary symmetric fixed shape reference

Condition	Mean Error (m)		RMSE (m)	
	NGA Asym	Arb Sym	NGA Asym	Arb Sym
15 m target, single calibration	-0.016	0.083	0.214	0.321
30 m target, single calibration	0.010	0.054	0.143	0.213
60 m target, single calibration	-0.025	-0.153	0.433	0.638
15 m target, average calibration	-0.011	0.079	0.108	0.192
30 m target, average calibration	-0.000	-0.029	0.157	0.214
60 m target, average calibration	-0.006	-0.094	0.248	0.272
0.3 m spaced target	-0.045	-0.055	0.485	0.596

Table 7 shows a comparison of the results achieved with the NGA asymmetric waveform and an arbitrarily chosen symmetric waveform 50 percent wider than the NGA symmetric reference used for previous comparisons. In all cases, The NGA asymmetric reference resulted in range accuracy 5 to 10 times better than that of the arbitrary symmetric waveform and displayed a fraction of the variance.

These results showed that the true value of the NOVAS algorithm was its ability to detect and characterize the pulse shape variation and generate average asymmetric and symmetric reference waveforms specific to that data set to use in a normalized correlation. The NGA reference waveforms improved the range accuracy of the normalized correlation range estimation technique.

5.5 Application to RPV Air-to-Air Refueling

Results of the two tests performed indicated that several concerns must be resolved before a system such as the ASC LADAR would be useful in providing relative ranging to an RPV with enough accuracy to allow close formation flight required for autonomous AAR. The largest of these concerns was the clock frequency variation which prevented achieving a consistent, accurate range estimation in real time or otherwise without two known-range targets in the LADAR FOV. The range error caused by dropped slices could also prevent a range solution at acceptable update rates if a temporal filter is required to detect and discard erroneous slice counts. Also a fac-

tor were individual pixel errors which resulted in significant range estimation errors. These are technical implementation factors for which solutions may reside in technological advancement or more robust application, allowing smaller range estimation errors to be found and solved.

In the absence of technical implementation factors, pulse shape effects may be the next largest source of range estimation errors. This thesis demonstrated that the laser pulse signals generated by a Q-switched laser cavity possessed variability in shape which affected the range estimation produced from a correlation algorithm using a non-optimized reference waveform. The ability to change the shape of the reference waveform gave the NOVAS algorithm the flexibility to adjust to the changing signal shape. It was shown that NOVAS retained the highest level of correlation despite varying pulse shapes.

Moreover, NOVAS was able to generate averaged fixed asymmetric and symmetric pulse shapes specific to a particular data set that improved the accuracy of range estimation and reduced the variation in those estimates over an arbitrarily chosen reference waveform. In the event pulse shape effects become the dominant error, NOVAS may hold the key to generating LADAR range estimates accurate enough to provide autonomous RPV air refueling capability in a GPS denied environment.

The following listings contain the Matlab® code used to generate the results for this thesis. The first listing is the code used to extract data from the LADAR output file called a sequence file with the suffix .seq. The Matlab® code in the first listing was not uniquely developed by the author who modified code that was previously used at the Air Force Institute of Technology. The remaining four Matlab® files were uniquely developed by the author for the purposes of this thesis.

The second listing contains the code that builds the reference matrices and their supporting structure which NOVAS uses to estimate the range and pulse shape of the data waveforms. Most of the complexity of the algorithm lies in building and indexing the reference matrices. NOVAS, which is contained in the third listing, is relatively straightforward. The fourth and fifth listings contain Matlab® scripts that call the NOVAS function to extract and organize the range and pulse width information in useable formats. It was from the outputs of these two files using a variety of reference matrices that results, analysis, and conclusions for this thesis were drawn.

Listing A.1: FLASH_READ_NO_TV.m

```
% This script extracts the output of the TPS ASC FLASH LADAR in
% the form of MM-DD-YEAR_HH_MM_SSXM.seq.  It was derived from
% a collection of scripts already in use at the Air Force
% Institute of Technology for reading LADAR output files.
% Major modifications include the ability to extract multiple
% files, a new output format, the ability to extract SULAR and
% stop mode files into the same output format, the circular
% shifting of the waves to place them in chronological order,
% the addition of a 2D image of maximum amplitude returns,
% and the identification of dead pixels.

% The script outputs and saves two structures, "Flash" and
% "Range" for each file. "Range" is simply a subset of "Flash"
% with only that portion of the data required for range esti-
```

```
% mation.  This script only saves the LADAR image and discards
% the TV image.

% INPUT FILE NAMES MANUALLY-----------------------------
% This vector of file names will be saved at the end of this
% script, overwriting any file with the name "file_names"
file_names=['11-09-2009_06_17_31PM';
            '11-09-2009_06_13_23PM';
            '11-09-2009_06_14_16PM'];

% INITIALIZE MATRICES--------------
mkr_pos=zeros(128,128);     % Position of the marker slice
max_return=zeros(128,128);  % Max value recorded by each pixel
waves=zeros(19,128,128);    % The recorded waveforms

for z=1:size(file_names,1)
filename=[file_names(z,:) '.seq'];

fid = fopen (filename,'rb');        % opens the .seq file
Flash.file=filename;                % Add filename to structure
filename=filename(12:19);           % Shortens filename

Flash.maxNumFrames = 300;

% BEGIN READING .SEQ HEADER INFORMATION-------
fseek(fid,0,'bof');
SeqHeader.DAC   = fread (fid, [30], 'float');    % Settings
SeqHeader.TempA = fread (fid, [128-30], 'uint');
SeqHeader.Camera_Type = fread(fid, 1, 'uint');
SeqHeader.Sular = fread(fid, 1, 'uint');         % 0=stop 1=SULAR
SeqHeader.Width = fread(fid, 1, 'uint');         % camera width
SeqHeader.Height = fread(fid, 1, 'uint');        % camera height
SeqHeader.Stop = fread(fid, 1, 'float');         % max range
SeqHeader.Hold = fread(fid, 1, 'float');
SeqHeader.No3D = fread(fid, 1, 'uint');
```

```
   SeqHeader.MarkerSlice = fread(fid, 1, 'uint');   % range to marker
51 SeqHeader.SelectIncrement = fread(fid, 1, 'uint');
   SeqHeader.StopIncrements = fread(fid, 1, 'uint');
   SeqHeader.SequenceNumber = fread(fid, 1, 'uint');
   SeqHeader.NumPictures = fread(fid, 1, 'uint');   % number of ...
      frames
   SeqHeader.NumSeconds = fread(fid, 1, 'float');
56 SeqHeader.DateTime = fread(fid, 1, 'uint');
   SeqHeader.ActualPicture = fread(fid, 1, 'uint');
   SeqHeader.VisRotate = fread(fid, 1, 'float');
   SeqHeader.ImageLast = fread(fid, 1, 'uint');
   SeqHeader.Motor0Pos = fread(fid, 1, 'float');
61 SeqHeader.Motor1Pos = fread(fid, 1, 'float');
   SeqHeader.PixelSlip = fread(fid, 1, 'uint');
   SeqHeader.Rotate3D = fread(fid, 1, 'uint');
   SeqHeader.IRPicture = fread(fid, 1, 'uint');
   SeqHeader.TempA1 = fread(fid, [12], 'uint');
66 SeqHeader.Visible_Pic = fread(fid, 1, 'uint');
   SeqHeader.JPEG = fread(fid, 1, 'uint');
   SeqHeader.myWidth = fread(fid, 1, 'uint');
   SeqHeader.myHeight = fread(fid, 1, 'uint');
   SeqHeader.format = fread(fid, 1, 'int');
71 SeqHeader.ImageSize = fread(fid, 1, 'uint');
   SeqHeader.QTable = fread(fid, 1, 'uint');
   SeqHeader.TempA2 = fread(fid, [20], 'uint');
   SeqHeader.INS = fread(fid, 1, 'uint');
   SeqHeader.INSSize = fread(fid, 1, 'uint');
76 SeqHeader.TempA3 = fread(fid, [10], 'uint');
   SeqHeader.Seq3D = fread(fid, 1, 'uint');
   SeqHeader.Size3D = fread(fid, 1, 'uint');
   SeqHeader.MaxGate = fread(fid, 1, 'float');
   SeqHeader.TempA4 = fread(fid, [128-76], 'uint');
81
   Flash.Header=SeqHeader;     % Load header into Flash structure
```

```
% PREPARE VARIABLES FOR EXTRACTING IMAGE DATA-----------
Flash.NumberOfFrames=Flash.Header.NumPictures;
width  = Flash.Header.Width;   % The width of each image
height = Flash.Header.Height;  % The height of each image

% BEGIN EXTRACTING IMAGE DATA ONE FRAME AT A TIME--------
 for currentFrame=1:Flash.NumberOfFrames
        disp(['Current Frame :',num2str(currentFrame)])

        %-----------------------------------------------------
         %Read in the visual camera (TV) data
         % 786432 words  (1+1024x768 16-Bit words)
         % This is not saved in the structure and is lost
         % at eachiteration for the sake of memory.
         vis=flipud((rot90(fread(fid, [1024 768], 'uint16'))));


        %-----------------------------------------------------
         % Read in the Hit Buffer
         % 65636 words  (4*128x128 16-Bit words)
         Flash.frame(currentFrame).HitBuffer1 = ...
          flipud((rot90(fread(fid, [width height], 'ushort'))));
         Flash.frame(currentFrame).HitBuffer2 = ...
          flipud((rot90(fread(fid, [width height], 'ushort'))));
         Flash.frame(currentFrame).HitBuffer3 = ...
          flipud((rot90(fread(fid, [width height], 'ushort'))));
         Flash.frame(currentFrame).HitBuffer4 = ...
          flipud((rot90(fread(fid, [width height], 'ushort'))));


        %-----------------------------------------------------
         % Filler Padding Vector
         %(DISCARDED)
         padding=fread(fid,128*88+1,'ushort');
```

```
%-------------------------------------------------------
 %Range to marker slice in number of slices units

 Flash.frame(currentFrame).MarkerRange=...
   flipud((rot90(fread(fid, [width height], 'ushort'))));


%-------------------------------------------------------
 % Read in the slices of 3-D data
 % 327680 words  (20*128*128 16-Bit words)
 % This includes the marker slice.

 for currentSlice=1:20
  Flash.frame(currentFrame).slice(currentSlice,:,:) = ...
   flipud((rot90(fread(fid, [width height], 'ushort'))));
 end

%-------------------------------------------------------
 % Skip a word of padding data at end of 3D data
 % 1 word (DISCARDED)
 padding = fread(fid, 1, 'uint16');


%-------------------------------------------------------
 % Read in the padding images
 % 32768 words (2*128*128 16-Bit words)
 Flash.frame(currentFrame).paddingIm1 = ...
  flipud((rot90(fread(fid, [width height], 'ushort'))));

 Flash.frame(currentFrame).paddingIm2 = ...
  flipud((rot90(fread(fid, [width height], 'uint16'))));


%-------------------------------------------------------
```

```
        % Skip 1022 words of padding data at end of the frame
        %
156     % The entire data block is 0xDE000 (909312 bytes,
        % 454656 shorts) in length.  Read in the remaining data
        % and throw away.  The remaining data is:
        % BlockSize - IR Size - DataSize - FooterSize short words
        % or 454656 - 76801 - 344064 - 32769 = 1022 16-Bit words.
161     padding = fread(fid, [1022], 'uint16');

    % SHIFT EACH WAVE INTO CHRONOLOGICAL ORDER---------------
    % ShiftS the marker slice to the beginning of each wave-
    % form and then discard the marker slice, leaving a 19
166 % slice long wave.  This preserves the corrupted slices
    % and places them in positions 1 and 19.

    cube=Flash.frame(currentFrame).slice;    % Get all waveforms
    mkr_rng=...
171   Flash.frame(currentFrame).MarkerRange; % Get marker ranges
    begin_range=mkr_rng+1;                   % Compensate for
                                             % discarded marker
    for i=1:128
        for j=1:128
176     wave1=squeeze(cube(:,i,j));

        [val pos]=min(wave1);          % Marker is lowest value
        mkr_pos(i,j)=pos;              % Load marker posit
        wave2=circshift(wave1,-pos+1); % Move marker to front
181     wave3=wave2(2:20);             % Discard marker slice
        waves(:,i,j)=wave3;            % Load adjusted waveform
        [val pos]=max(max(wave3));     % Find max value of wave
        max_return(i,j)=val;           % 128x128 of max values

186     end
    end
    DEAD_val=max(max(max_return)); % Dead pixel=huge amplitude
```

```
        ignore=max_return;              % Prepare "ignore" matrix
        ignore=ignore==DEAD_val;        % Boolean IDs dead pixels
191     max_return(ignore)=0;           % Dead pixels=zero amplitude

        % BUILD RANGE STRUCTURE-----------------
        Range.Frame(currentFrame).waves=waves;
        Range.Frame(currentFrame).begin_range=begin_range;
196     Range.Frame(currentFrame).max_return=max_return;

    end  % end frame loop

        % SAVE OUTPUT -------------------------
201         filename
            eval(['save ' filename ' Flash'])
            name=strcat('Range',filename)
            eval(['save ' name ' Range'])
            save file_names file_names
206         clear
            load file_names

end  % end file loop
```

Listing A.2: BUILD_REFERENCE.m

```
1
% This script builds a reference waveform structure for use
% with NOVAS.m with the inputs BUILD_REFERENCE(num_samples)
% where num_samples is the number ofrange subsamples between
% recorded slices.
6
% The output is a structure containing all the pertinent data for
% the particular set of reference waveforms.  Most important is
% the .matrix element, which is a 17x17xM array of reference
% waveforms.  The first dimension (layer) is the position (of 17
```

```
% "slices") of the peak data point, and is the starting point for
% analysis.  The algorithm then scans the reference waveform
% across a section of the frame, which is contained in the
% respective layer of the R_precise element of the structure.
% Except for data with peaks at either end of the frame, each
% scanned range section is centered on the frame containing the
% peak recorded amplitude.  For a single-surface return, this
% will always capture the actual peak of the return laser pulse.


% USER DEFINED REFERENCE PULSE PARAMETERS----------------------

num_samples_per=10;      % The number of range increments
                         % between slices

sigmaL=[0.6130 0.7152 0.8173 0.9195 1.0216 1.1238 1.2260 1.3281...
        1.4303 1.5325 1.6346 1.7368 1.8390 1.9411 2.0433 2.1455];
sigmaR=[0.9195 1.0216 1.1238 1.2260 1.3281 1.4303 1.5325 1.6346...
        1.7368 1.8390 1.9411 2.0433 2.1455 2.2476 2.3498 2.4520];
         % The available left and right half-pulse widths

scan_points=5;     % This must be an odd number.  It defines the
                   % number of slices scanned by the reference
                   % waveform.  3 scan points makes NOVAS run
                   % faster, but occasionally misses the peak.

% CALCULATED PARAMETERS----------------------------------------
num_samples=num_samples_per*16+1;  % total range increments
num_scan_steps=num_samples_per*(scan_points-1)+1;
                                   % sub-scan increments
num_pulse_shapes=(length(sigmaL)*length(sigmaR));
num_waveforms=num_scan_steps*num_pulse_shapes;

% RANGE GATE PARAMETERS----------------------------------------
```

```
R=[1:17]';                          % 17 slices per frame
R_sub=[1:1/num_samples_per:17]';    % Subsampled frame

% MATRIX PRE-ALLOCATION---------------------------------------

R_precise=zeros(17,num_samples_per*(scan_points-1)+1);
Ref=zeros(17,17,num_waveforms);
Ref_norm=zeros(17,17,num_waveforms);

% BUILD 17 SUB-SCAN RANGE VALUES------------------------------

for j=1:17
    if j<scan_points-1
        R_precise(j,:)=linspace(R(1),R(scan_points),...
            num_scan_steps);
    elseif j>17-scan_points+1
        R_precise(j,:)=linspace(R(17-scan_points+1),...
            R(17),num_scan_steps);
    else
        R_precise(j,:)=linspace(R(j-(scan_points-1)/2),...
            R(j+(scan_points-1)/2),num_scan_steps);
    end
end

% THIS EXTRACTS 17 POINTS FROM A SUB-SAMPLED GAUSSIAN----------

conversion_index=(0:16)*num_samples_per+1;

%  Reference Matrix for Correlation----------------------------

% Indexes defined below are used specifically in the NOVAS
% algorithm for identification of range, sigma_l and sigma_r.
% The order of permutation is sigmaL, sigmaR,range
% steps.  For example, if there are 2 possible choices for
% sigmaL, 3 possible choices for sigmaR, and num_scan_steps=5,
```

```
% then the 30 columns of the reference matrix will be indexed
% as follows:
%
%        SigmaL -> L1                L2
%        SigmaR -> R1    R2    R3    R1    R2    R3
% 5 Range Steps -> 12345 12345 12345 12345 12345 12345
%     Waveforms -> ||||| ||||| ||||| ||||| ||||| |||||
%    wave index -> 1234..                        ...30

for j=1:17  % j defines the layer of the matrix corresponding
            % to the slice of the data with the peak amplitude.
     j
     wave_index=1;

for l=1:length(sigmaL)  % l defines the left half pulse width
for k=1:length(sigmaR)  % k defines the right half pulse width
for i=1:num_scan_steps  % i defines position within the sub_scan

% BUILD INDIVIDUAL WAVEFORMS--------------------------------

     R_ref=R_precise(j,i);  % Assigns peak range of the reference

     Left_side=exp((-(R_sub-R_ref).^2)./(2*sigmaL(l)*sigmaL(l)));
     Right_side=exp((-(R_sub-R_ref).^2)./(2*sigmaR(k)*sigmaR(k)));
        % Complete, equal-amplitude, non-normalized, symmetric
        % sub-sampled Gaussian waves to use as building blocks

 if j≤ceil(scan_points/2) % When peak amplitude is at left end

    if i==1   % if "firewalled left" only use right side
       Ref_ful=Right_side(conversion_index);
    else      % if not, use transision to join left and right
              % to build piecewise Gaussian
       trans_point=floor(i/num_samples_per+1);
       Ref_ful=[Left_side(conversion_index(1:trans_point));...
```

```
116             Right_side(conversion_index((trans_point+1):17))];
        end


    elseif j≥17-floor(scan_points/2) % When peak is at right end
121
        if i==num_scan_steps  % if "firewalled right" only use left
            Ref_ful=Left_side(conversion_index);
        else      % if not, use transision to join left and right
                  % to build piecewise Gaussian
126         trans_point=floor(i/num_samples_per+(17-scan_points+1));
            Ref_ful=[Left_side(conversion_index(1:trans_point));...
                  Right_side(conversion_index((trans_point+1):17))];
        end


131
    else  % when in the "middle zones" no "firewall" occurs
          % Join left and right to build piecewise Gaussian

        if i==1
136       Ref_ful=...
           [Left_side(conversion_index(1:j-ceil(scan_points/2)));
            Right_side(conversion_index(j-floor(scan_points/2):17))];
        elseif i==num_scan_steps
            Ref_ful=...
141          [Left_side(conversion_index(1:j+floor(scan_points/2)-1));
             Right_side(conversion_index(j+floor(scan_points/2):17))];
        else
            trans_point=...
                j-floor(scan_points/2)+ceil(i/num_samples_per);
146         Ref_ful=...
                [Left_side(conversion_index(1:(trans_point-1)));
                 Right_side(conversion_index(trans_point:17))];
        end
    end
```

```
151  %  NORMALIZE REFERENCE WAVE ----------------------
     M_ref=mean(Ref_ful);
     SD_ref=std(Ref_ful,1);
     Ref_n=(Ref_ful-M_ref)/SD_ref;

156  Ref_norm(j,:,wave_index)=Ref_n;  % Assign waveform to the matrix
     wave_index=wave_index+1;         % Increment waveform index
    end
    end
    end
161 end

    % FORMULATE OUTPUT-------------------------------

    Reference.matrix=Ref_norm;
166 Reference.R_precise=R_precise;
    Reference.num_samples=num_samples;
    Reference.scan_points=scan_points;
    Reference.num_scan_steps=num_scan_steps;
    Reference.sigmaL=sigmaL;
171 Reference.sigmaR=sigmaR;

    lowest=num2str(min(sigmaL));
    highest=num2str(max(sigmaR));
    name=['REF_17_'  num2str(num_samples_per) 'per' ...
176     num2str(scan_points) '_' lowest(1) '_' highest(1)];

    save(name,'Reference')
```

Listing A.3: NOVAS.m

```
    function [X]=NOVAS(data, Reference)
  2 % a normalized non-interpolated slide-and multiply correlation
    % that inputs a 1x17 row vector and reference structure created
```

```
% by BUILD_REFERENCE and outputs the estimated position of the
% peak of the input data waveform (between slice 1 and slice 17)
% as well as an estimate of the left and right pulsewidth.

% PULL PARTS OF REFERENCE STRUCTURE INTO USEABLE MATRICES------
matrix=Reference.matrix;            % 17x17xM reference waveforms
Range=Reference.R_precise;          % Associated range estimates
num_scan_steps=Reference.num_scan_steps; % Number of range steps
sigmaL=Reference.sigmaL;            % Left pulse width values
sigmaR=Reference.sigmaR;            % Right pulse width values
num_sigmaR=length(sigmaR);

% FIND THE LAYER OF THE REFERENCE MATRICES
% CENTERED ON THE PEAK OF THE DATA WAVEFORM------------------
[val peak]=max(data);               % Slice with max amplitude
matrix=squeeze(matrix(peak,:,:));  % Get corresponding layer
Range=squeeze(Range(peak,:));      % Get corresponding layer

% NORMALIZE THE DATA-----------------------------------------
m_data=mean(data);                  % Get the mean
SD_data=std(data,1);                % Get the standard deviation
data_norm=(data-(m_data*ones(1,17)))./SD_data;  % Normalize

% PERFORM CORRELATION----------------------------------------
corr=data_norm*matrix;
[val index]=max(corr);   % "index" points to one of "M" wave-
                         % forms with the highest correlation
match=squeeze(matrix(:,index));    % isolates the match waveform

%--MAP "INDEX" TO CORRECT PARAMETER ESTIMATIONS
% For this it is important to know how the reference matrix
% is formed. The order of permutation is sigmaL, sigmaR,range
% steps.  For example, if there are 2 possible choices for
% sigmaL, 3 possible choices for sigmaR, and num_scan_steps=5,
```

```
% then the 30 columns of the reference matrix will be indexed
% as follows:
%
%        SigmaL -> L1                 L2
%        SigmaR -> R1    R2    R3     R1    R2    R3
% 5 Range Steps -> 12345 12345 12345 12345 12345 12345
%     Waveforms -> ||||| ||||| ||||| ||||| ||||| |||||
%         index -> 1234..                         ...30
%    left_index -> 11111 11........1 22222 22........2
%        index2 -> 12345 12345 ...            ...12345
%right_index_sub-> 12345 6789.....15 12345 6789.......15
%   right_index -> 11111 22222 33333 11111 22222 33333

% RANGE IS FOUND WITH INDEX2----------------------------
index2=rem(index,num_scan_steps);
if index2==0
    index2=num_scan_steps;
end
est=Range(index2);

% LEFT HALF PULSE WIDTH IS FOUND WITH LEFT_INDEX-------
left_index=ceil(index/(num_sigmaR*num_scan_steps));
sigma_left=sigmaL(left_index);

% RIGHT HALF PULSE WIDTH IS FOUND WITH RIGHT_INDEX_SUB
% AND RIGHT_INDEX--------------------------------------
right_index_sub=rem(index,(num_sigmaR*num_scan_steps));
if right_index_sub==0
    right_index_sub=num_sigmaR*num_scan_steps;
end
right_index=ceil(right_index_sub/num_scan_steps);


sigma_right=sigmaR(right_index);
```

```
X=[[index; est; sigma_left; sigma_right; peak; val;...
    zeros(size([7:17]))'],[match]];     %Outputs 17x2 matrix
  % ^ Placeholder zeros
```

Listing A.4: RANGE_EXTRACT_GTEST1.m

```
% This script was used to extract useful portions from
% the first ground test.  This extracts useful
% information from the "Range" structure output from the
% FLASH_READ_NO_TV script file and saves them as a structure
%% Initialize Variables-------------------------------------
sigR_all=[];      % Right pulse widths
sigL_all=[];      % Left pulse widths
slice_all=[];     % Target position in frame
slice_tot_all=[]; % Target range (in slices)
match_all=[];     % matched reference waveforms
norm_all=[];      % normalized data waveforms
raw_all=[];       % non-normalized data waveforms

save sigR_all 'sigR_all'
save sigL_all 'sigL_all'
save slice_all 'slice_all'
save slice_tot_all 'slice_tot_all'
save match_all 'match_all';
save norm_all 'norm_all';
save raw_all 'raw_all';

%% Extract Information--------------------------------------
for file_num=1:76
    file_num

    % LOAD THE REQUIRED FILES----------------------------------
    load filenames   % contains portions of desired file namems
    load pixels      % contains 19 pixel coordinates [Vert Horiz]
```

```
load('Refsep_17_10per_5scan_halfns_15_60.mat')
                % load the reference structure

file=filenames(file_num, :);
load(['Range' file '.mat']); % load a single Rangexxx.mat
                             % created by FLASH_READ_NO_TV

% CREATE USEABLE STRUCTURE-------------------------------
TEMP=Range.Frame;
num_frames=length(TEMP);
corr_waveform=Reference.matrix;

for i=1:num_frames

  % CREATE USEABLE MATRICES-------------------------------
  cube=TEMP(i).waves;                  % data waveforms
  begin_range=TEMP(i).begin_range;     % marker slice ranges
  max_return(i,:,:)=TEMP(i).max_return;% 2-D image

for pix_num=1:19

  Vert=pixels(pix_num,1);              % Pixel coordinates
  Horiz=pixels(pix_num,2);

  data=squeeze(cube(:,Vert,Horiz))';  % get single waveform
  data=data(2:18);                    % cut corrupted slices

  m_data=mean(data);                  % Normalize data
  SD_data=std(data,1);
  data_norm(i,pix_num,:)=(data-(m_data*ones(1,17)))./SD_data;

  range_start=begin_range(Vert,Horiz)+1;
                    % get starting slice count and compensate
                    % for discarded first slice
```

```
        temp=NOVAS(data,Reference);     % Run NOVAS

             match_index(i,pix_num) = temp(1);
67                 slice(i,pix_num) = temp(2);
             sigma_l_est(i,pix_num) = temp(3);
             sigma_r_est(i,pix_num) = temp(4);
               slice_tot(i,pix_num) = slice(i,pix_num)+range_start;
        match_waveform(i,pix_num,:) = temp(:,2);
72                 raw(i,pix_num,:) = data;

    end
    end
    % LOAD RESULTS INTO TEMPORARY STRUCTURE---------------------
77  TEMP2.raw=raw;
    TEMP2.slice=slice;
    TEMP2.sigL=sigma_l_est;
    TEMP2.sigR=sigma_r_est;
    TEMP2.match=match_waveform;
82  TEMP2.slice_tot=slice_tot;
    TEMP2.norm=data_norm;
    TEMP2.max_return=max_return;

    % SAVE RESULTS BY FILE--------------------------------------
87  eval(['extr_' num2str(file_num) '=TEMP2;']);
    eval(['save extr_' num2str(file_num)...
          ' extr_' num2str(file_num) ';']);

    % SAVE RESULTS BY DATA TYPE---------------------------------
92  load sigR_all
    sigR_all(file_num,:,:)=sigma_r_est;
    save sigR_all 'sigR_all';
    clear sigR_all

97  load sigL_all
    sigL_all(file_num,:,:)=sigma_l_est;
```

```
        save sigL_all 'sigL_all';
        clear sigL_all

102     load slice_all
        slice_all(file_num,:,:)=slice;
        save slice_all 'slice_all';
        clear slice_all

107     load slice_tot_all
        slice_tot_all(file_num,:,:)=slice_tot;
        save slice_tot_all 'slice_tot_all';
        clear slice_tot_all

112     load raw_all
        raw_all(file_num,:,:,:)=raw;
        save raw_all 'raw_all';
        clear raw_all

117     load norm_all
        norm_all(file_num,:,:,:)=data_norm;
        save norm_all 'norm_all';
        clear norm_all

122     load match_all
        match_all(file_num,:,:,:)=match_waveform;
        save match_all 'match_all';
        clear match_all

127     clear

    end

    %% Locate bad waveforms
132 % This cell picks out waveforms with maximum amplitudes greater
    % than 3000 (saturated pulses) or less than 1000 (low SNR).
```

```
% It creates a boolean that can then be applied to any of the
% "results by data type" matrices created above to isolate
% results from "good" waveforms only.

load raw_all
for file_num=1:76
  for frame=1:36
    for pix=1:19

      if max(squeeze(raw_all(file_num,frame,pix,:)))>3000;
          bad_wave_bool(file_num,frame,pix)=1==1;
      else bad_wave_bool(file_num,frame,pix)=1≠1;
      end  % IF SATURATED THEN TRUE, IF NOT, THEN FALSE

      if max(squeeze(raw_all(file_num,frame,pix,:)))<1000;
          bad_wave_bool(file_num,frame,pix)=1==1;
      end  % IF LOW SNR THEN TRUE

    end
  end
end

save bad_wave_bool 'bad_wave_bool'

%% Locate non-returns
% Pixels that do not have a target return indicate a range
% measurement of 27,000 slices.  This boolean helps pickout
% such pixels in the same manner as "bad_wave_bool".

load slice_tot_all
non_return_bool=slice_tot_all>20000;
save non_return_bool 'non_return_bool';
```

Listing A.5: RANGE_EXTRACT_GTEST2.m

```
% This script was used to extract useful portions from
% the second ground test.  This extracts useful
% information from the "Range" structure output from the
% FLASH_READ_NO_TV script file and saves them as a structure
% with whatever name you specify in line 37 below.

for file=1:20
    % MANUALLY INSERT THE DESIRED FILENAMES HERE
    % (CHANGE 1:21 ABOVE AS REQUIRED TO INCLUDE ALL FILES)-----
    filenames=['Range06_49_32.mat';'Range06_49_18.mat';
               'Range06_47_46.mat';'Range06_46_27.mat';
               'Range06_45_54.mat';'Range05_50_35.mat';
               'Range06_11_26.mat';'Range06_13_23.mat';
               'Range06_14_16.mat';'Range06_17_31.mat';
               'Range06_43_47.mat';'Range06_17_42.mat';
               'Range06_43_35.mat';'Range06_25_03.mat';
               'Range06_40_23.mat';'Range06_25_50.mat';
               'Range06_39_51.mat';'Range06_30_50.mat';
               'Range06_39_02.mat';'Range06_31_22.mat'];

    load('REF_17_10per5_1_2.mat')        % load Reference

    load(filenames(file,:))              % load requested file
    name=['wide_' filenames(file,9:13)]  % name=wide_mm_ss

    num_frames=length(Range.Frame);

    for i=1:num_frames
        i
        % INITIALIZE MATRICES---------------
        raw=zeros(128,128,17);   % non-normalized data waveforms
        norm=zeros(128,128,17);  % normalized data waveforms
        match=zeros(128,128,17); % matched reference waveforms
        slice=zeros(128,128);    % target position in frame
        slice_tot=zeros(128,128);% total slices to target
```

```
sigL=zeros(128,128);      % left pulse width
sigR=zeros(128,128);      % right pulse width

% CREATE USEFUL WAVEFORM MATRIX--------
cube=Range.Frame(i).waves;

% EXTRACT USEFUL PIXELS FROM MIG VERTICAL TAIL----------
for Vert=56:63
  for Horiz=21:33

  data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
  data=data(2:18);                    % Discard the
                                      % corrupted slices
  m_data=mean(data);                  % normalize data
  SD_data=std(data,1);
  norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

  slice_start=(Range.Frame(i).begin_range)'+1;
                              % Get marker slice range
                              % and compensate for
                              % discarded first slice

  temp=NOVAS(data,Reference); % Run NOVAS

      raw(Vert,Horiz,:) = data;
      slice(Vert,Horiz) = temp(2,1);
       sigL(Vert,Horiz) = temp(3,1);
       sigR(Vert,Horiz) = temp(4,1);
    match(Vert,Horiz,:) = temp(:,2);
 slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                               +slice_start(Vert,Horiz);
  end
end

% EXTRACT USEFUL PIXELS FROM TARGET BOARD-------------
```

```
for Vert=73:80
  for Horiz=73:76
  data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
  data=data(2:18);                    % Discard the
                                      % corrupted slices
  m_data=mean(data);                  % normalize data
  SD_data=std(data,1);
  norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

  slice_start=(Range.Frame(i).begin_range)'+1;
                                % Get marker slice range
                                % and compensate for
                                % discarded first slice

  temp=NOVAS(data,Reference); % Run NOVAS

      raw(Vert,Horiz,:) = data;
      slice(Vert,Horiz) = temp(2,1);
       sigL(Vert,Horiz) = temp(3,1);
       sigR(Vert,Horiz) = temp(4,1);
    match(Vert,Horiz,:) = temp(:,2);
 slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                              +slice_start(Vert,Horiz);
  end
end

% EXTRACT USEFUL PIXELS FROM TARGET BOARD PART 2--------
for Vert=74:85
  for Horiz=60:64
  data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
  data=data(2:18);                    % Discard the
                                      % corrupted slices
  m_data=mean(data);                  % normalize data
  SD_data=std(data,1);
  norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;
```

```
107           slice_start=(Range.Frame(i).begin_range)'+1;
                                % Get marker slice range
                                % and compensate for
                                % discarded first slice

112           temp=NOVAS(data,Reference); % Run NOVAS

                raw(Vert,Horiz,:) = data;
                slice(Vert,Horiz) = temp(2,1);
                 sigL(Vert,Horiz) = temp(3,1);
117              sigR(Vert,Horiz) = temp(4,1);
               match(Vert,Horiz,:) = temp(:,2);
             slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                                        +slice_start(Vert,Horiz);
              end
122         end

           % EXTRACT USEFUL PIXELS FROM TARGET BOARD PART 3---------
           for Vert=74:83
             for Horiz=43:48
127          data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
             data=data(2:18);                    % Discard the
                                                 % corrupted slices
             m_data=mean(data);                  % normalize data
             SD_data=std(data,1);
132          norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

             slice_start=(Range.Frame(i).begin_range)'+1;
                                % Get marker slice range
                                % and compensate for
137                             % discarded first slice

             temp=NOVAS(data,Reference); % Run NOVAS
```

```
            raw(Vert,Horiz,:) = data;
142         slice(Vert,Horiz) = temp(2,1);
             sigL(Vert,Horiz) = temp(3,1);
             sigR(Vert,Horiz) = temp(4,1);
          match(Vert,Horiz,:) = temp(:,2);
       slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
147                                 +slice_start(Vert,Horiz);
        end
      end

      % EXTRACT USEFUL PIXELS FROM THE VAN--------------------
152   for Vert=54:67
        for Horiz=6:21
        data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
        data=data(2:18);                    % Discard the
                                            % corrupted slices
157     m_data=mean(data);                  % normalize data
        SD_data=std(data,1);
        norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

        slice_start=(Range.Frame(i).begin_range)'+1;
162                             % Get marker slice range
                                % and compensate for
                                % discarded first slice

        temp=NOVAS(data,Reference); % Run NOVAS
167
            raw(Vert,Horiz,:) = data;
            slice(Vert,Horiz) = temp(2,1);
             sigL(Vert,Horiz) = temp(3,1);
             sigR(Vert,Horiz) = temp(4,1);
172       match(Vert,Horiz,:) = temp(:,2);
       slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                                    +slice_start(Vert,Horiz);
        end
```

```
end

% EXTRACT USEFUL PIXELS FROM THE VAN PART 2-------------
for Vert=53:57
  for Horiz=37:52
  data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
  data=data(2:18);                    % Discard the
                                      % corrupted slices
  m_data=mean(data);                  % normalize data
  SD_data=std(data,1);
  norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

  slice_start=(Range.Frame(i).begin_range)'+1;
                              % Get marker slice range
                              % and compensate for
                              % discarded first slice

  temp=NOVAS(data,Reference); % Run NOVAS

      raw(Vert,Horiz,:) = data;
      slice(Vert,Horiz) = temp(2,1);
       sigL(Vert,Horiz) = temp(3,1);
       sigR(Vert,Horiz) = temp(4,1);
    match(Vert,Horiz,:) = temp(:,2);
 slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                                +slice_start(Vert,Horiz);
  end
end

% 100 FOOT TARGET-------------------------------
Vert=89;
Horiz=94;
  data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
  data=data(2:18);                    % Discard the
                                      % corrupted slices
```

```
            m_data=mean(data);                  % normalize data
212         SD_data=std(data,1);
            norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

            slice_start=(Range.Frame(i).begin_range)'+1;
                                    % Get marker slice range
217                                 % and compensate for
                                    % discarded first slice

            temp=NOVAS(data,Reference); % Run NOVAS

222             raw(Vert,Horiz,:) = data;
                slice(Vert,Horiz) = temp(2,1);
                 sigL(Vert,Horiz) = temp(3,1);
                 sigR(Vert,Horiz) = temp(4,1);
               match(Vert,Horiz,:) = temp(:,2);
227         slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                                        +slice_start(Vert,Horiz);
        % 100 FOOT TARGET-----------------------------------
        Vert=88;
        Horiz=94;
232       data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
          data=data(2:18);                    % Discard the
                                              % corrupted slices
          m_data=mean(data);                  % normalize data
          SD_data=std(data,1);
237       norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

          slice_start=(Range.Frame(i).begin_range)'+1;
                                    % Get marker slice range
                                    % and compensate for
242                                 % discarded first slice

          temp=NOVAS(data,Reference); % Run NOVAS
```

```
            raw(Vert,Horiz,:) = data;
247         slice(Vert,Horiz) = temp(2,1);
             sigL(Vert,Horiz) = temp(3,1);
             sigR(Vert,Horiz) = temp(4,1);
          match(Vert,Horiz,:) = temp(:,2);
       slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
252                                 +slice_start(Vert,Horiz);
      % 150 FOOT TARGET--------------------------------------
      Vert=79;
      Horiz=87;
        data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
257     data=data(2:18);                    % Discard the
                                            % corrupted slices
        m_data=mean(data);                  % normalize data
        SD_data=std(data,1);
        norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;
262
        slice_start=(Range.Frame(i).begin_range)'+1;
                                  % Get marker slice range
                                  % and compensate for
                                  % discarded first slice
267
        temp=NOVAS(data,Reference); % Run NOVAS

            raw(Vert,Horiz,:) = data;
            slice(Vert,Horiz) = temp(2,1);
272          sigL(Vert,Horiz) = temp(3,1);
             sigR(Vert,Horiz) = temp(4,1);
          match(Vert,Horiz,:) = temp(:,2);
       slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                                    +slice_start(Vert,Horiz);
277   % 150 FOOT TARGET--------------------------------------
      Vert=81;
      Horiz=86;
          data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
```

```
    data=data(2:18);                    % Discard the
282                                     % corrupted slices
    m_data=mean(data);                  % normalize data
    SD_data=std(data,1);
    norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

287     slice_start=(Range.Frame(i).begin_range)'+1;
                                % Get marker slice range
                                % and compensate for
                                % discarded first slice

292     temp=NOVAS(data,Reference); % Run NOVAS

        raw(Vert,Horiz,:) = data;
        slice(Vert,Horiz) = temp(2,1);
         sigL(Vert,Horiz) = temp(3,1);
297      sigR(Vert,Horiz) = temp(4,1);
       match(Vert,Horiz,:) = temp(:,2);
 slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                                  +slice_start(Vert,Horiz);
% 50 FOOT TARGET--------------------------------------
302 Vert=104;
Horiz=120;
    data=squeeze(cube(:,Horiz,Vert))';  % Get waveform
    data=data(2:18);                    % Discard the
                                        % corrupted slices
307 m_data=mean(data);                  % normalize data
    SD_data=std(data,1);
    norm(Vert,Horiz,:)=(data-(m_data*ones(1,17)))./SD_data;

    slice_start=(Range.Frame(i).begin_range)'+1;
312                             % Get marker slice range
                                % and compensate for
                                % discarded first slice
```

```
            temp=NOVAS(data,Reference); % Run NOVAS
317
                raw(Vert,Horiz,:) = data;
                slice(Vert,Horiz) = temp(2,1);
                 sigL(Vert,Horiz) = temp(3,1);
                 sigR(Vert,Horiz) = temp(4,1);
322           match(Vert,Horiz,:) = temp(:,2);
          slice_tot(Vert, Horiz) = slice(Vert,Horiz)...
                                          +slice_start(Vert,Horiz);
      % ENTER ALL VALUES INTO THE STRUCTURE-----------------------

327   eval([name '.frame(' num2str(i) ').raw=raw;']);
      eval([name '.frame(' num2str(i) ').norm=norm;']);
      eval([name '.frame(' num2str(i) ').match=match;']);
      eval([name '.frame(' num2str(i) ').slice=slice;']);
      eval([name '.frame(' num2str(i) ').slice_tot=slice_tot;']);
332   eval([name '.frame(' num2str(i) ').sigL=sigL;']);
      eval([name '.frame(' num2str(i) ').sigR=sigR;']);
      eval([name '.frame(' num2str(i) ...
                  ').max_return=Range.Frame(i).max_return;']);
       end   % End the loop on frames
337
       % SAVE THE STRUCTURE----------------------------------
       eval(['save ' name ' ' name])
       clear
   end  % End the loop on files
```

Bibliography

1. *TPS Operator's Manual, ASC FLASH LADAR Video Camera (FLVC).* October 2009.

2. Burris, Charles R. *Detection and Ranging of Obscured Targets in 3-D LADAR Data Using Estimation Theory.* Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), March 2006.

3. Cain, Stephen. "Flash Light Detection and Ranging Range Accuracy Limits for Returns from Single Opaque Surfaces Via Cramer-Rao Bounds". *Applied Optics*, 45(24):6154–6162, 2006.

4. Jelalian, Albert V. *Laser Radar Systems.* Artech House, Boston, MA, 1992.

5. Liu, Jingang. "Modeling Pulse Shape of Q-Switched Lasers". *IEEE Journal of Quantum Electronics*, 37(7):888–896, 2001.

6. Poor, H. Vincent. *An Introduction to Signal Detection and Estimation, Second Edition.* Springer, New York, NY, 1994.

7. R. Stettner, H. Bailey and R. Richmond. "Eye-Safe Laser Radar 3D Imaging". *Proc. SPIE Vol. 5412, Laser Radar Technology and Applications IX*, 111–116, 2004.

8. Samuel D. Stearns, Ruth A. David. *Signal Processing Algorithms.* Prentice Hall, Inc, Englewood Cliffs, New Jersey 07632, 1988.

9. Seal, Michael D. *Nonlinear (Time Variant) Response in an Avalanche Photodiode Array Based Laser Detection and Ranging System.* Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright Patterson AFB, OH, March 2007.

10. Siegman, Anthony E. *Lasers.* University Science Books, Sausalito, CA, 1986.

11. Spinelli, Christopher J. *Development and Testing of a High-Speed Real-Time Kinematic Precise DGPS Positioning System Between Two Aircraft.* Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), September 2006.

12. Wilcox, Rand R. *Applying Contemporary Statistical Techniques.* Academic Press, 525 B Street, Suite 1900, San Diego, CA 92101, 2003.

Vita

Major Brian T. Deas was born at Barksdale Air Force Base (AFB). He graduated from Southwest High School in Fort Worth, Texas in 1995. He graduated from the United States Air Force Academy in 1999 with a Bachelor of Science Degree in Physics.

His first assignment in the United States Air Force (USAF) was to Sheppard AFB, Texas where he attended Euro-Nato Joint Jet Pilot Training (ENJJPT). After graduating ENJJPT in 2000, Major Deas was selected for F-16 training at Luke AFB, Arizona where he was assigned to the 62^{nd} Fighter Squadron. Major Deas was reassigned in 2001 to the 522^{nd} Fighter Squadron at Cannon AFB, New Mexico where he flew F-16C and F-16CJ aircraft in support of OPERATION NOBLE EAGLE and deployed in support of OPERATION IRAQI FREEDOM. In 2004 Major Deas was again reassigned, this time to the 13^{th} Fighter Squadron at Misawa Air Base, Japan where he served as a Flight Commander and instructor pilot.

In 2007 Major Deas attended the Air Force Institute of Technology in conjunction with his selection for the USAF Test Pilot School (TPS). He graduated from TPS in December 2009 as an Experimental Test Pilot and will receive his Masters Degree in Electrical Engineering in March 2010. His next assignment will be at the 416^{th} Flight Test Squadron at Edwards AFB, where he will perform developmental testing on the the F-16.

Permanent address: On File

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
25-03-2010	Master's Thesis	June 2007-March 2010

4. TITLE AND SUBTITLE

Pulse Shape Correlation for Laser Detection and Ranging (LADAR)

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Deas, Brian,Major, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765 DSN: 785-3636

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/10-07

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
INTENTIONALLY LEFT BLANK

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

An un-biased range estimation algorithm was developed that countered errors caused by pulse shape effects in laser radar (LADAR) systems. The Advanced Scientific Concepts FLASH LADAR created three-dimensional images by measuring return intensity and time of flight for a single laser pulse. A range estimate was made by estimating the return time of flight from the peak amplitude of the return laser pulse. Range error estimates occurred when the shape of the measured return pulse deviated from the expected pulse shape.

A normalized variable shaping correlation algorithm called NOVAS performed the correlation between the recorded waveforms and a robust set of Gaussian-based reference waveforms covering a wide range of pulse shapes and asymmetries. While NOVAS did not generate the most accurate range estimates due to noise sensitivity, it was able to produce an estimate of the average asymmetric pulse shape. The NOVAS generated average (NGA) asymmetric pulse shape produced a more accurate average range estimate than the NGA symmetric pulse shape in six of seven trials and produced a more accurate average range estimate with less variance than an arbitrarily chosen symmetric reference waveform in all of seven trials.

15. SUBJECT TERMS
Laser Detection and Ranging; LADAR; LIDAR; Range Estimation; Pulse Shape; Normalized Correlation; Q-switched Laser Pulses; 3D Imaging; Variable Shaping Estimator; Pulse Shape Correlation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	ABSTRACT	c. THIS PAGE			Dr. Stephen Cain
U	U	U	UU	121	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565; email: Stephen.cain@afit.edu